

Escuela Politécnica Superior

19
20

Trabajo fin de grado

Desarrollo de skills accesibles para asistentes de voz comerciales



Alberto Moreno Brasero

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C/ Francisco Tomás y Valiente nº 11

**UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Desarrollo de skills accesibles para asistentes de
voz comerciales**

Autor: Alberto Moreno Brasero

Tutor: Javier Gómez Escribano

Ponente: Germán Montoro Manrique

junio 2020

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

DERECHOS RESERVADOS

© 3 de Noviembre de 2017 por UNIVERSIDAD AUTÓNOMA DE MADRID
Francisco Tomás y Valiente, nº 1
Madrid, 28049
Spain

Alberto Moreno Brasero

Desarrollo de skills accesibles para asistentes de voz comerciales

Alberto Moreno Brasero

C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

A mis padres por aguantarme y a mi tutor por guiarme en este proyecto.

*El futuro mostrará los resultados
y juzgará a cada uno de acuerdo a sus logros.*

Nikola Tesla

AGRADECIMIENTOS

En un primer lugar (dada su relevancia) debo agradecer a Javier Gómez su labor como tutor del TFG. Gracias a su ayuda he podido llevar a cabo la realización de este trabajo de principio a fin. Agradezco que me haya sabido guiar en el desarrollo del mismo dado que ha habido bastantes ocasiones en las que me encontraba perdido y no sabía por donde salir. Dentro de ese ámbito él ha sabido hacer su trabajo perfectamente sin extralimitarse en el ejercicio de sus funciones de guía. En ese sentido, me ha dejado libertad para que sea yo el que trabajase en el proyecto realmente y así poder dejar mi esencia en el mismo. También quiero agradecerle el seguimiento semanal de mi trabajo que demuestra el interés que tiene en sus alumnos del TFG que permite que no haya sorpresas indeseadas. Por último, quería agradecerle que me acercase al mundo del desarrollo de soluciones informáticas para personas con diversidad intelectual dado que era un área que desconocía completamente. Gracias a esto, ahora mismo veo las dificultades que tiene y lo necesario que es para poder ayudar a esta parte de la población.

En un segundo lugar debo agradecer a mis padres y a mi hermano el apoyo que me han dado. En especial, a mi padre dado que por su posición de antiguo universitario me ha dado consejos sobre cómo acometer el TFG en el aspecto de los plazos y algunos aspectos de redacción del mismo.

En tercer lugar quería agradecer a Cristina su apoyo emocional durante el presente curso en el que estaba trabajando en este TFG y las asignaturas que me restaban para completar la carrera. También le agradezco que se prestase a ayudarme a elegir temas para el TFG relacionados con la diversidad funcional intelectual.

Por último, quería agradecer a aquellos compañeros con los cuáles me he cruzado durante los 4 años de esta carrera que me han ayudado (y a los que he ayudado) a resolver dudas sobre conceptos de las asignaturas de la misma. También les agradezco el tenerme en cuenta para ámbitos fuera de la universidad.

RESUMEN

Los asistentes virtuales (así como los altavoces inteligentes) están tomando una gran relevancia en el mercado mundial. Estos asistentes virtuales utilizan como método de interacción principal el lenguaje natural. Unido a lo anterior existe una parte de la población que tiene dificultades a la hora de expresarse de esta forma que son las personas con diversidad funcional intelectual. Sabiendo esto, comprobamos en la literatura científica y en la vida real que no existen casi desarrollos que permitan hacer esta tecnología más accesible para personas con problemas cognitivos.

Este trabajo parte de la problemática anterior y pretende realizar un desarrollo de una capa de accesibilidad sobre la funcionalidad de un temporizador en un asistente virtual que permite ser personalizada para que aquella persona con diversidad funcional intelectual (en específico Trastornos del Espectro Autista, TEA) entienda correctamente los estímulos que le llegan y reconozca sin ningún tipo de ayuda que se trata de un temporizador. Los estímulos del temporizador son la luz y el sonido. Estos estímulos se pueden configurar por la persona de apoyo para que el modo final elegido sea familiar a la persona con diversidad funcional intelectual. El asistente de voz utilizado es Alexa dentro del altavoz inteligente llamado Amazon Echo. Así, con la ayuda de esta tecnología que nos hace la vida más fácil a todas las personas que la utilizamos podría tener un gran potencial para personas con diversidad funcional intelectual a la hora de ayudarles a realizar tareas de la vida real.

El diseño de la respuesta del temporizador y la extracción de requisitos del sistema se ha realizado siguiendo las recomendaciones de expertos en TEA y se ha creado una arquitectura que permite aumentar la funcionalidad del proyecto de una manera fácil para personas con conocimientos en informática añadiendo al proyecto la funcionalidad que se desea implementar sin modificar ninguna de las partes que existen en la arquitectura.

Este documento recoge: una breve contextualización de las necesidades de las personas con diversidad funcional intelectual (y, en específico de las personas con TEA), la importancia de los asistentes virtuales en el mercado actual, un análisis de los requisitos de la funcionalidad del temporizador, el diseño de la arquitectura del proyecto, la implementación y conexión de los diferentes componentes y las pruebas realizadas a cada componente de la arquitectura por separado y en su conjunto.

PALABRAS CLAVE

asistentes virtuales, trastornos del espectro autista, diversidad funcional intelectual, Alexa, TEA

ABSTRACT

Nowadays virtual assistants are important on the world market (like smart speakers). The assistants can be used only with the voice. Also, the people with intellectual functional diversity can't speak properly. This issue makes difficult the communication between these people and virtual assistants. After a research we can observe there aren't solutions for people with intellectual functional diversity to help them to communicate with this technology.

This work started to solve the problems explained before. The solution is an accessibility layer over the timer feature in a virtual assistant. The user can customize timer's output with the purpose of the recognition of timer's output by a person with intellectual functional diversity (specifically Autism Spectrum Disorder, ASD) without any type of aid (in this case the user is the guardian of the ASD person). The timer response can be configured with light and sound. This response can be set by the support person of the ASD person to make timer's output familiar to the ASD person. The virtual assistant used is Alexa inside an Amazon Echo speaker. Using this technology and making it more accessible to people with ASD can help to make their lives easier.

The timer response and the system requirements were designed according to the results of meetings with experts in ASD. The architecture was designed in order to add new features to the project on a easier way. The architecture was designed to add new features without modifying the code written and the configuration made before.

The reader can see on this document a compilation of the needs of people with intellectual functional diversity (specifically people with ASD), the virtual assistant's importance on the current market, an analysis of the timer's features, the design of the solution, the implementation of the solution, the connection of the different components and the tests of the project architecture (tests were made on every component separately and on the entire architecture).

KEYWORDS

virtual assistants, autism spectrum disorder , intellectual functional diversity, Alexa

ÍNDICE

| | | |
|----------|---|-----------|
| 1 | Introducción | 1 |
| 1.1 | Motivación | 1 |
| 1.2 | Objetivos | 2 |
| 1.3 | Estructura de la memoria | 2 |
| 2 | Estado del arte | 3 |
| 2.1 | Diversidad funcional intelectual | 3 |
| 2.1.1 | Trastornos del Espectro Autista | 3 |
| 2.2 | Asistentes de voz comerciales | 4 |
| 2.2.1 | Amazon Alexa. | 6 |
| 2.2.2 | Bixby. | 6 |
| 2.2.3 | Cortana. | 6 |
| 2.2.4 | Google Assistant. | 7 |
| 2.2.5 | Mycroft. | 7 |
| 2.2.6 | Siri. | 7 |
| 2.3 | Situación del mercado de los altavoces inteligentes | 8 |
| 2.4 | Altavoces inteligentes en investigación | 9 |
| 2.5 | Conclusiones | 11 |
| 3 | Diseño | 13 |
| 3.1 | Necesidades de las personas con TEA | 13 |
| 3.2 | Requisitos del proyecto | 14 |
| 3.2.1 | Requisitos funcionales | 14 |
| 3.2.2 | Requisitos no funcionales | 14 |
| 3.3 | Diseño de la solución propuesta | 15 |
| 4 | Implementación | 17 |
| 4.1 | Alexa Skill | 20 |
| 4.1.1 | Modelo de interacción. | 20 |
| 4.1.2 | Función Lambda en AWS. | 22 |
| 4.2 | Altavoz Echo | 23 |
| 4.2.1 | Instalación de la skill en el Amazon Echo. | 23 |
| 4.3 | Servidor Raspberry Pi | 24 |
| 4.3.1 | Programa servidor. | 24 |

| | |
|---|-----------|
| 4.3.2 Base de datos | 26 |
| 4.3.3 Conexión con el dispositivo Amazon Echo. | 28 |
| 4.3.4 Comunicación con la skill. | 29 |
| 4.4 Philips Hue | 29 |
| 4.4.1 Arquitectura interna de las Philips Hue. | 30 |
| 4.4.2 Conexión con el servidor. | 31 |
| 5 Pruebas | 33 |
| 5.1 Pruebas unitarias | 33 |
| 5.1.1 Luz | 33 |
| 5.1.2 Sonido | 34 |
| 5.1.3 Servidor | 34 |
| 5.1.4 Skill | 35 |
| 5.2 Pruebas de integración | 35 |
| 5.2.1 Luz | 35 |
| 5.2.2 Sonido | 36 |
| 5.2.3 Servidor | 36 |
| 5.2.4 Skill | 36 |
| 6 Conclusiones y trabajo futuro | 37 |
| 6.1 Conclusiones | 37 |
| 6.2 Trabajo futuro | 38 |
| Bibliografía | 40 |
| Definiciones | 41 |
| Acrónimos | 43 |

LISTAS

Lista de figuras

| | | |
|-----|--|----|
| 2.1 | Encuesta sobre las preguntas realizadas a altavoces inteligentes por parte de los usuarios. | 5 |
| 2.2 | Ranking de los asistentes virtuales de voz más utilizados en España en 2019. | 5 |
| 2.3 | Previsión del número de asistentes virtuales en uso a nivel mundial de 2019 a 2023. . . | 8 |
| 2.4 | Distribución porcentual de los envíos de altavoces inteligentes a nivel mundial en 2017 y 2018, por fabricante. | 9 |
| 2.5 | Marcas líderes de altavoces inteligentes. | 10 |
| 2.6 | Distribución porcentual de los usuarios de altavoces inteligentes en el hogar en España en 2019, por marca. | 10 |
| 3.1 | | 16 |
| 4.1 | Diseño de la arquitectura y flujo de ejecución. | 19 |
| 4.2 | Configuración de la respuesta del temporizador. | 25 |
| 4.3 | Tabla de valores de configuración. | 26 |
| 4.4 | Configuración actual de la respuesta del temporizador. | 27 |
| 4.5 | Adaptador Bluetooth. | 28 |
| 4.6 | Elementos de la arquitectura de Philips Hue. | 30 |

Lista de tablas

| | | |
|-----|--|---|
| 2.1 | Comparativa de los diferentes asistentes virtuales. | 8 |
|-----|--|---|

INTRODUCCIÓN

1.1. Motivación

Los asistentes de voz han sufrido una creciente demanda durante los 3 últimos años y cada día están más presentes en la vida cotidiana. Prueba de ello es que cada vez más servicios de atención al cliente utilizan asistentes de voz. También cada vez es más común utilizarlos debido a que se encuentran en los *smartphones*, ordenadores y en las casas en forma de altavoces inteligentes.

Asimismo, estos asistentes por voz son muy poco accesibles para personas con Trastornos del Espectro Autista (TEA) y les cuesta utilizarlos. Estos tienen ventajas relativas a que nos proporcionan experiencias de comunicación diferentes a las que existen ya que nos podemos comunicar mediante la voz. Esta forma de interacción es interesante para personas con problemas visuales o de movilidad. El problema reside es que esta forma de interacción excluye a las personas con TEA dado que tienen que utilizar un lenguaje natural y estas personas tienen un déficit de desarrollo en el ámbito de la comunicación verbal. Esto provoca que exista una dificultad de comunicación de estas personas con los asistentes virtuales. Se observa que actualmente no se han desarrollado soluciones para los principales asistentes de voz que permitan acceder a estas tecnologías a personas con TEA.

Este trabajo toma como base la investigación realizada por Leonardo Torres en su Trabajo Fin de Máster [1] que se trataba de una investigación sobre el uso de asistentes virtuales por personas con diversidad funcional intelectual. En esta investigación se encontraron las dificultades que presentan las personas con TEA a la hora de programar tareas en un asistente virtual.

En este trabajo se expone el desarrollo de una capa de accesibilidad para personas con TEA en una funcionalidad específica de un asistente virtual. La funcionalidad desarrollada es un temporizador personalizable que permite elegir entre diferentes estímulos para que la persona con TEA reconozca que es un temporizador cuando esta se active. La finalidad de este trabajo se basa en probar si los asistentes de voz están preparados para ser accesibles a personas con necesidades específicas.

La principal motivación para realizar este TFG ha sido que dado que los asistentes por voz son una tecnología muy demandada actualmente y que cuenta con pocas opciones de accesibilidad para

personas con diversidad funcional es necesario comenzar este trabajo de adaptación de este tipo de tecnologías a un público con necesidades específicas. El paso que se da en esta dirección debe producirse para que este público no se quede atrás en el uso de esta tecnología.

Otra motivación que hubo detrás del presente trabajo ha sido probar si el desarrollo de funcionalidad de terceros en un asistente virtual comercial está lo suficientemente avanzado para adaptar estas tecnologías a personas con TEA y así hacer estos productos más personalizables y accesibles.

1.2. Objetivos

Los objetivos principales de este trabajo son los siguientes:

- Desarrollar un programa para un asistente de voz comercial que facilite el acceso a estos servicios para personas con diversidad funcional intelectual. Así se aumenta la oferta de soluciones para estos usuarios.
- Permitir que las personas con TEA puedan entender y comunicarse cómodamente con un asistente de voz comercial y puedan ordenar tareas a los asistentes virtuales con facilidad.
- Facilitar que la respuesta del temporizador pueda ser modificada de una forma bastante flexible para que la persona de apoyo pueda modificarse el estímulo a uno que la persona con TEA pueda reconocer fácilmente como temporizador.
- Estudiar las limitaciones que tienen los asistentes virtuales existente para satisfacer las necesidades de los usuarios con diversidad funcional intelectual. Así verificamos si los asistentes de voz virtuales están preparados para atender a las necesidades de personas con TEA.
- Estudiar las limitaciones de las personas con TEA a la hora de utilizar los asistentes virtuales.

1.3. Estructura de la memoria

El documento está dividido en diferentes puntos. El motivo de esta organización es proporcionar al lector una visión completa del proyecto, desde los antecedentes del mismo y el contexto en el que se encuadra hasta los detalles del desarrollo completo del mismo (así como las dificultades encontradas).

En la **introducción** se pone en contexto al lector sobre aquello que se pretende realizar en el proyecto, se explica la motivación que mueve la creación del proyecto y se establecen los objetivos finales del proyecto. En el **estado del arte** se explican conceptos importantes, las tecnologías más relevantes para el proyecto y se realiza una comparativa entre estas tecnologías. En el **diseño** se realiza una explicación detallada de la creación del diseño de la solución a emplear y la elección de los componentes. En la **implementación** se establece una explicación de la realización completa del proyecto partiendo del diseño y se explica la conexión entre los diferentes elementos de la arquitectura de la solución planteada así como el desarrollo detallado de cada parte de la arquitectura. En las **conclusiones y trabajo futuro** se realiza un resumen de los conocimientos adquiridos en la realización del proyecto y del trabajo futuro que se podría producir para ampliar y mejorar el proyecto realizado.

ESTADO DEL ARTE

2.1. Diversidad funcional intelectual

La diversidad funcional intelectual se define como un déficit general en las funciones cognitivas que surgen durante la infancia. También requiere de un funcionamiento social deteriorado cuya consecuencia es una reducida independencia personal o una necesidad de cuidado o protección especial [2].

Según la OMS la diversidad funcional intelectual es una capacidad muy reducida para entender información compleja o nueva y de aprender nuevas habilidades. Estas premisas generan que existe una capacidad reducida para ser independiente y este cambio se produce en la infancia, lo que provoca un efecto duradero en el desarrollo intelectual del individuo.

Esta condición no depende exclusivamente de la salud del niño si no que también influye de una manera importante el ambiente en el que convive en el sentido de si este apoya la participación e inclusión total del niño en la sociedad.

El uso de este término incluye a personas con autismo y a niños que han sido rechazados por sus familias que adquieren como consecuencia retrasos en el desarrollo y problemas psicológicos [3].

2.1.1. Trastornos del Espectro Autista

El concepto de TEA se refiere a todos los fenómenos que se encuentran asociados al **autismo**. Estos fenómenos están dentro de un grupo de trastornos neurológicos del desarrollo [4]. El concepto ya mencionado se encuentra dentro de la clasificación del DSM-V [5]. En este manual se explican los síntomas más importantes que permiten reconocer el TEA en individuos. Las características más importantes de las personas con TEA son:

- Dificultades en la comunicación.
- Dificultades en la interacción social.
- Patrones repetitivos de comportamiento, intereses y actividades. Estos se manifiestan en los siguientes síntomas:
 - Obsesiones.

- Rebeldía.
- Frustración.

El autismo se trata de un espectro porque no todos los individuos diagnosticados tienen los mismos síntomas ni en el mismo grado. Por este motivo se divide en diferentes categorías en las cuáles el individuo con autismo es más compatible con una de ellas según los síntomas que presente. Para situar una persona con autismo en una zona u otra del espectro se debe atender al desarrollo que tiene en sus capacidades. Una persona se encuentra en la parte baja del espectro cuando sus capacidades comunicativas y sociales son limitadas o inexistentes. Por otro lado, una persona en la zona alta del espectro puede comunicarse con fluidez y hay veces que sus síntomas son imperceptibles a simple vista. Aun así es muy difícil definir categorías dentro del espectro autista [2].

Al ser tan heterogéneos los síntomas de los TEA tienen como consecuencia que el diagnóstico del autismo sea complejo y costoso [6]. Existen ocasiones en las cuales aunque los síntomas estén presentes en la infancia, de estos no se extrae un diagnóstico completo hasta la adolescencia o la etapa adulta. Los síntomas de estos trastornos suelen ser calificados por los expertos como **dificultad para el control de las acciones** o **disfunción ejecutiva** [2, 7].

2.2. Asistentes de voz comerciales

Los asistentes virtuales son programas software que permiten que las comunicaciones entre un sistema informático y una persona sean más fáciles a la hora de que las personas realicen tareas en estos sistemas. En este caso, los asistentes virtuales utilizan la voz como medio de comunicación para que el usuario planifique una tarea y para que el sistema informático resuelva la respuesta al usuario sobre lo que éste le haya ordenado.

Las principales funcionalidades de estos asistentes virtuales según los propios usuarios se pueden ver en la figura 2.1 que es una encuesta de la empresa **comScore** realizada en Estados Unidos muestra que el 60 % de los usuarios de altavoces inteligentes lo utiliza para realizar preguntas generales, el 57 % para realizar consultas sobre el tiempo atmosférico y el 54 % para escuchar música del mismo altavoz. [8]

En cuanto al entorno español (que es principalmente al que nos dirigimos) se obtiene que, en 2019, Siri es el asistente de voz más utilizado por los españoles (86,2 %) muy lejos de sus competidores, seguido del asistente de Google (11,6 %) y, en el último lugar del podio, se puede encontrar a Alexa (9,8 %) con una cuota de usuarios más modesta como se puede ver en la figura 2.2. [9]

El funcionamiento de los asistentes de voz más populares suele ser bastante parecido entre ellos: estos realizan una escucha continua de los sonidos del exterior y se activan cuando se pronuncia la frase de activación. En algunos casos también permiten una entrada de texto por parte del usuario.

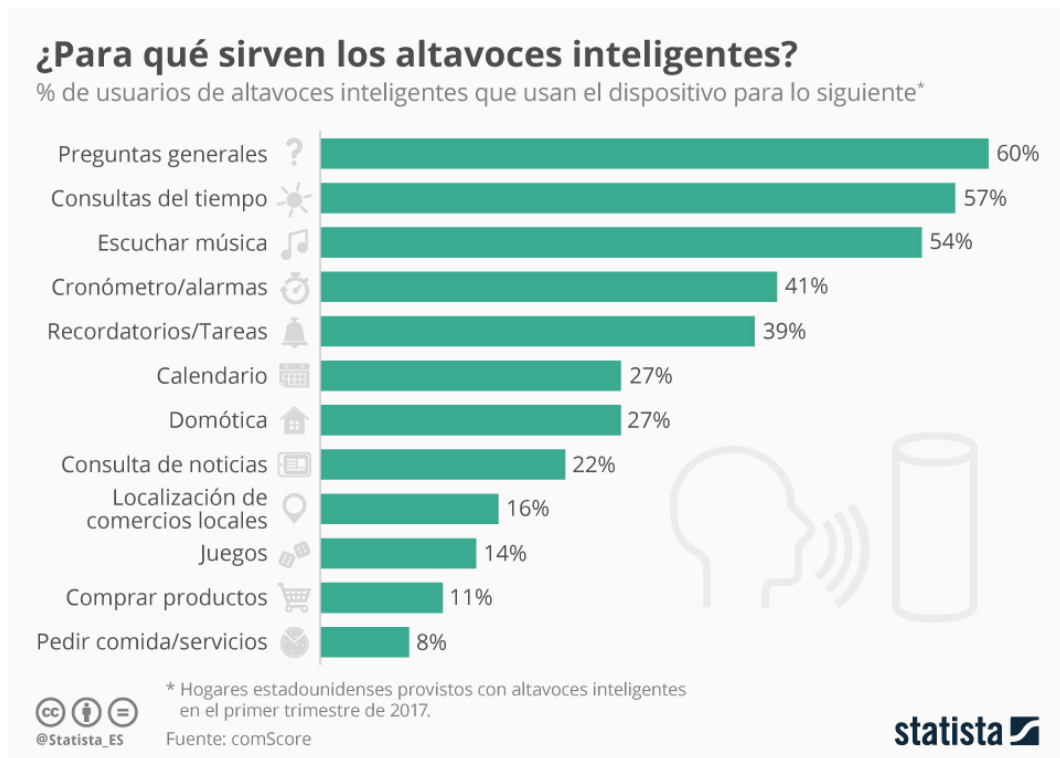


Figura 2.1: Encuesta sobre las preguntas realizadas a altavoces inteligentes por parte de los usuarios. [8]

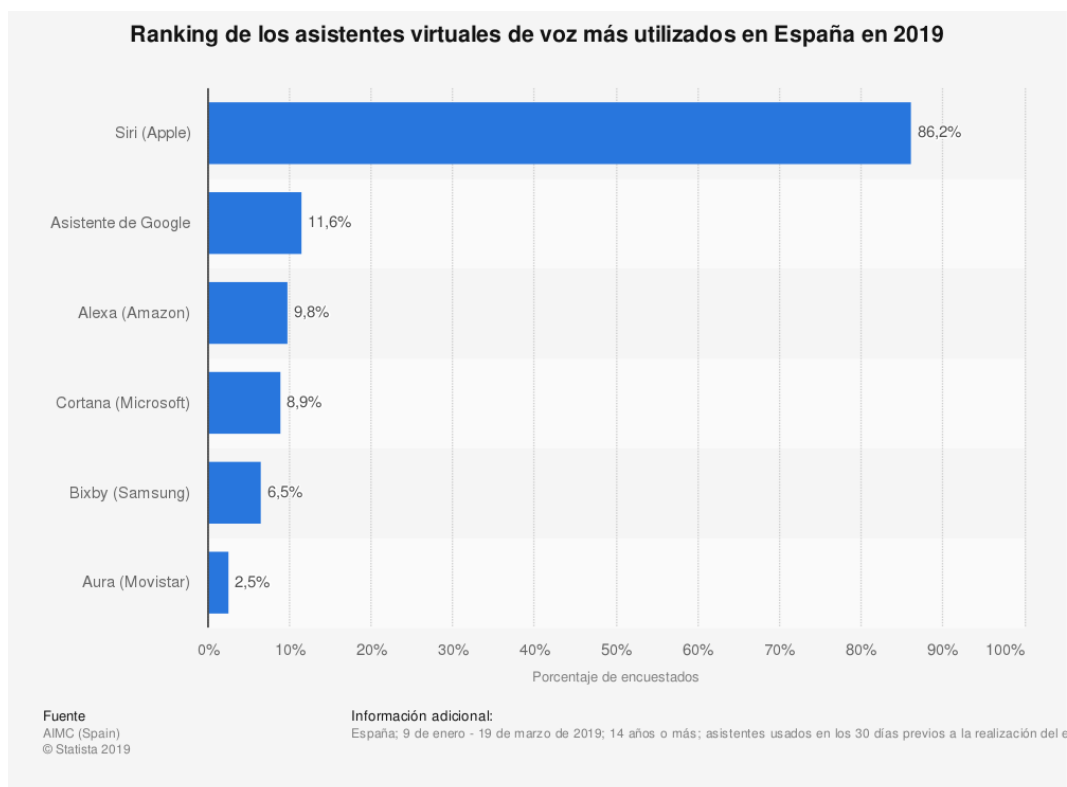


Figura 2.2: Ranking de los asistentes virtuales de voz más utilizados en España en 2019. [9]

Se ha realizado un estudio de los diferentes asistentes virtuales y se han elegido aquellos que son más populares del mercado (a excepción de **Mycroft** que se ha incluido por la relevancia que es que se trata del único asistente virtual de software libre). A continuación se habla brevemente de cada uno de los que se han elegido.

2.2.1. Amazon Alexa.

Se trata de un asistente virtual desarrollado por Amazon cuyo lanzamiento fue realizado en el año 2014. Fue utilizado por primera vez en los dispositivos Amazon Echo desarrollados y comercializados por Amazon.

Este asistente virtual puede controlar diferentes dispositivos inteligentes compatibles con él (focos, interfonos, cámaras de videovigilancia...). La funcionalidad de este asistente se puede extender instalando skills desde la aplicación Alexa. También se pueden crear rutinas para automatizar algunos procesos con un comando de voz, hora o ubicación.

La página web del proyecto es: <https://developer.amazon.com/es-ES/alexa> (Accedido: 02-06-2020).

2.2.2. Bixby.

Este es un asistente virtual desarrollado por la compañía Samsung. Su lanzamiento está fechado en el 21 de Abril de 2017 integrado en el Samsung Galaxy S8. Este asistente tiene tres partes: Bixby Voice, Bixby Vision y Bixby Home. La parte que interesa es la de Bixby Voice que es el asistente virtual por voz. Mediante el Samsung Developer Kit se pueden desarrollar aplicaciones que aumenten la funcionalidad de Bixby desde el Bixby Marketplace. También se pueden crear rutinas para automatizar algunos procesos como con Alexa.

La página web del proyecto es: <https://www.samsung.com/us/explore/bixby/> (Accedido: 02-06-2020).

2.2.3. Cortana.

Se trata de un asistente virtual creado por Microsoft para todos los sistemas desarrollados para Microsoft y en el futuro para Alexa. Su lanzamiento se realizó el 2 de abril de 2014. Este asistente puede responder preguntas mediante el motor de búsqueda de Bing, reconocer la voz o el texto y poder establecer recordatorios. Cortana tiene el soporte de IFTTT y otras habilidades desarrolladas por Microsoft para hardware domótico externo.

La página web del proyecto es: <https://www.microsoft.com/en-us/cortana/> (Accedi-

do: 02-06-2020).

2.2.4. Google Assistant.

Este asistente ha sido desarrollado por Google. El lanzamiento se realizó en mayo de 2016. Este se ha implementado en la mayoría de los dispositivos Android y el dispositivo Google Home. Los usuarios interactúan mediante voz natural pero también pueden interactuar mediante la entrada de texto. Este asistente puede mostrar la información de la cuenta Google del usuario, buscar en Internet, programar eventos y alarmas y cambiar la configuración hardware del dispositivo. En este asistente no se pueden instalar habilidades extra (o aplicaciones) que extiendan la funcionalidad del mismo.

La página web del proyecto es: <https://assistant.google.com/> (Accedido: 02-06-2020).

2.2.5. Mycroft.

Este es un asistente virtual de software libre y código abierto bajo la licencia Apache 2.0. Su fecha de lanzamiento fue el 4 de abril de 2016. Está pensado para sistemas operativos basados en Linux (Android, Linux y Raspbian). Por otro lado, también existe hardware específico (unos altavoces inteligentes) que tienen este sistema integrado. Este asistente tiene la funcionalidad de hacer búsquedas por Internet, poner alarmas y recordatorios, crear eventos, responder preguntas y controlar cosas del software y hardware del dispositivo. El código al ser libre es completamente personalizable. Por ello, se puede modificar y adaptar para solventar un problema en concreto. Para aumentar la funcionalidad del asistente virtual, Mycroft cuenta con una tienda de skills para la cual puede desarrollar cualquier persona para integrar soluciones de terceros en este asistente. Estas skills se desarrollan en el lenguaje de programación Python.

La página web del proyecto es: <https://mycroft.ai/> (Accedido: 02-06-2020).

2.2.6. Siri.

Se trata de una inteligencia artificial con funciones de asistente personal para los dispositivos de Apple. El lanzamiento de Siri fue realizado en 2011. Esta inteligencia artificial utiliza el procesamiento del lenguaje natural para responder preguntas, hacer solicitudes a servicios web y hacer recomendaciones. A esta inteligencia artificial no se le puede añadir funcionalidad extra por parte de habilidades o aplicaciones de terceros como en Alexa y tampoco se pueden realizar desarrollos independientes de funcionalidad.

La página web del proyecto es: <https://www.apple.com/siri/> (Accedido: 02-06-2020).

En la tabla 2.1 se puede ver una comparativa entre los diferentes asistentes virtuales.

| Asistente virtual | Desarrollo de terceros | Código abierto | Multiplataforma | Altavoz inteligente |
|-------------------|------------------------|----------------|----------------------|--------------------------|
| Alexa | Sí | No | Sí | Amazon Echo |
| Bixby | Sí | No | Sí | Samsung Galaxy Home Mini |
| Cortana | Sólo IFTT | No | No | Invoke |
| Google Assistant | No | No | Android y Google SOs | Google Home |
| Mycroft | Sí | Sí | Sí | Mycroft Mark |
| Siri | No | No | No | Homepod |

Tabla 2.1: Comparativa de los diferentes asistentes virtuales.

2.3. Situación del mercado de los altavoces inteligentes

La situación actual del mercado de asistentes de voz para el público en general se centra en dispositivos compuestos de un altavoz con conexión a Internet y un micrófono con el que se puede comunicar para que efectúe tareas de domótica u otras tareas como consultar información en Internet, responder preguntas y tareas en un ámbito más genérico. Este mercado se ha demostrado que tiene bastante éxito y se prevé que va a seguir creciendo con fuerza en un futuro como vemos en la figura 2.3.

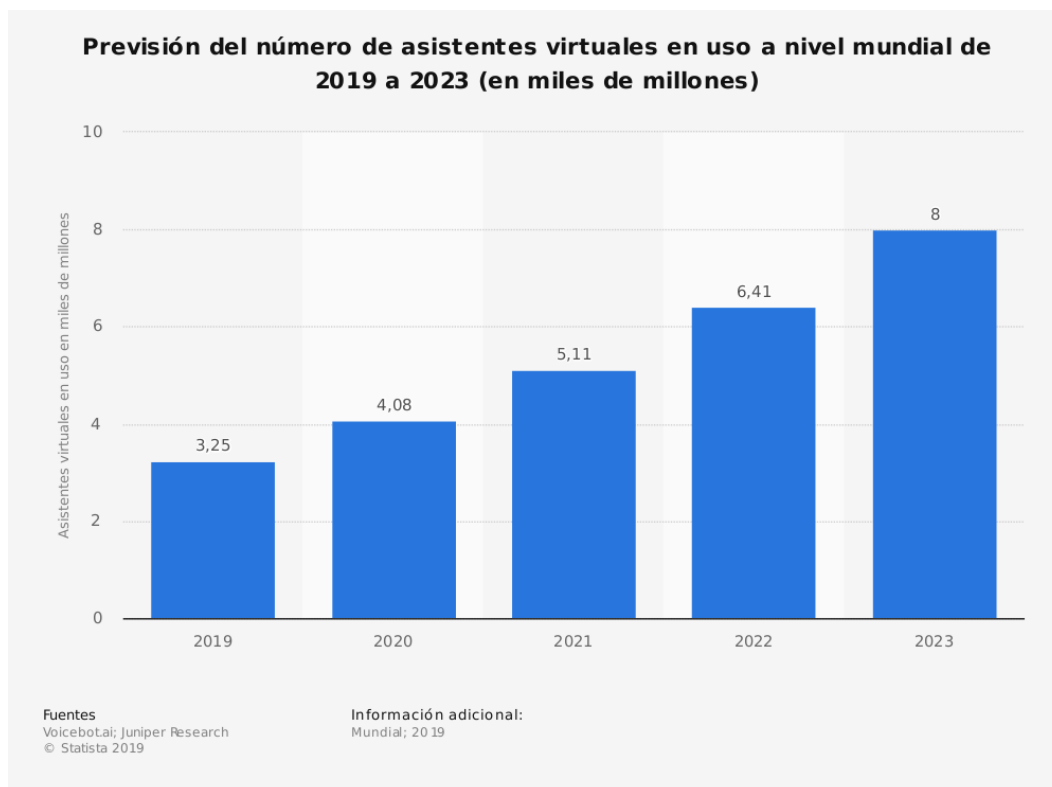


Figura 2.3: Previsión del número de asistentes virtuales en uso a nivel mundial de 2019 a 2023. [10]

Hay principalmente dos compañías que lideran el sector de altavoces inteligentes según se observa en la figura 2.4: Amazon y Google. Aún así, desde que el Amazon Echo (el dispositivo con Alexa de Amazon) fue lanzado al mercado en 2014 han surgido multitud de altavoces inteligentes con su correspondiente asistente de voz y aún así se sigue viendo que Amazon encabeza el número de compras de altavoces inteligentes como se puede observar en la figura 2.5.

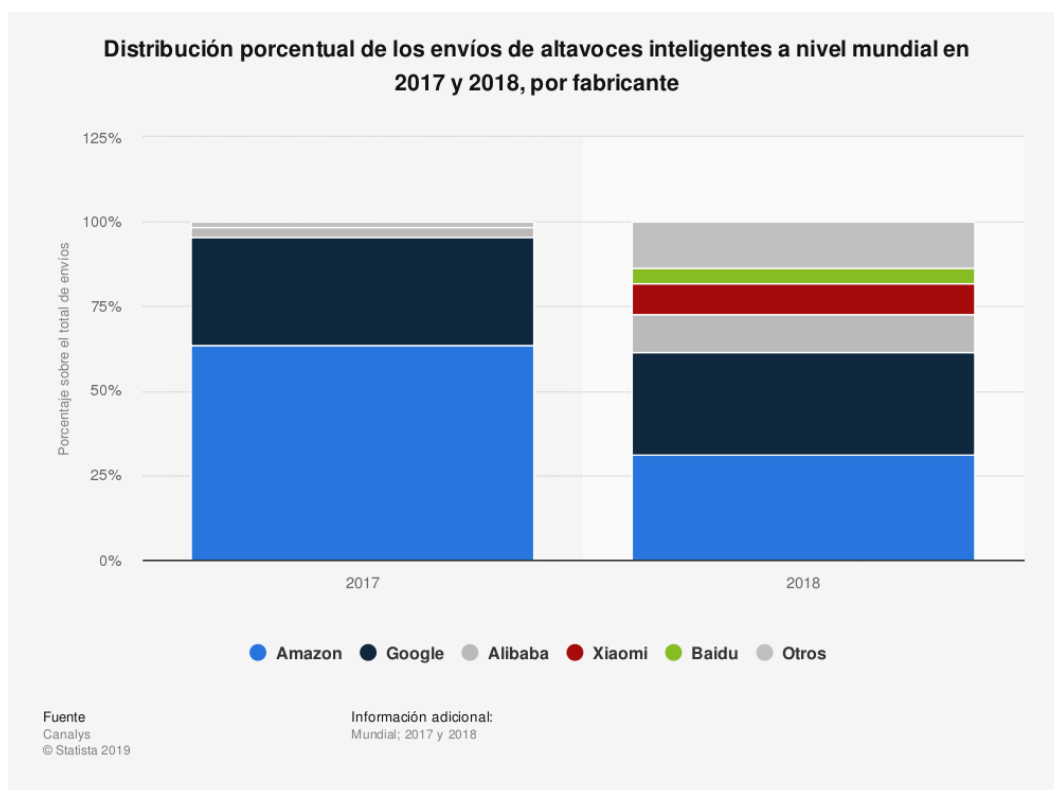


Figura 2.4: Distribución porcentual de los envíos de altavoces inteligentes a nivel mundial en 2017 y 2018, por fabricante. [11]

En cuanto a los altavoces inteligentes que más cuota de mercado tienen España (que es el principal mercado al que se dirige esta solución), en el año 2019, está en primera posición el altavoz Google Home de Google (35 %). En segunda posición, seguido muy de cerca de la primera están los Amazon Echo con Alexa de Amazon (33,6 %). En última posición, se encuentra el Home Pod de Apple con una cuota de mercado bastante baja (7 %). Estos datos se pueden observar en la figura 2.6. [13]

2.4. Altavoces inteligentes en investigación

Los altavoces inteligentes son una buena herramienta para utilizarla en entornos de aprendizaje de personas con TEA dado que se ha demostrado en un estudio que los dispositivos de salida de audio (Voice Output Communication Aid, VOCA) con un discurso sintetizado aumentan las habilidades comunicativas de una persona con TEA y permite que haya un mayor entendimiento con las otras

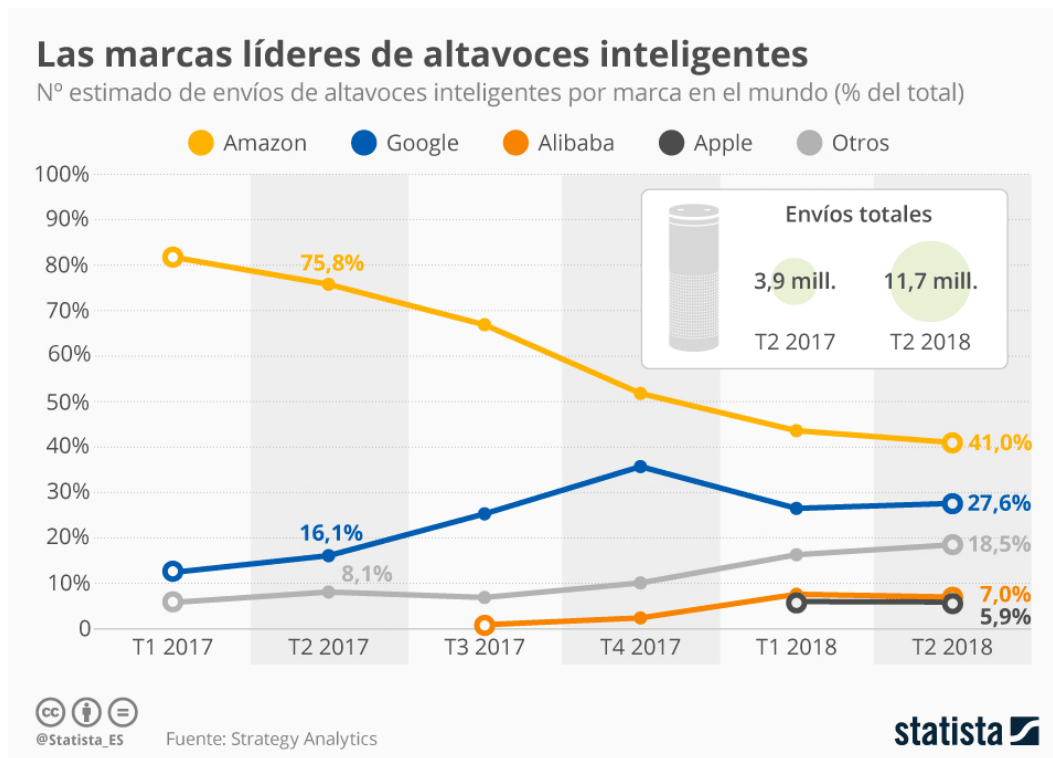


Figura 2.5: Marcas líderes de altavoces inteligentes. [12]

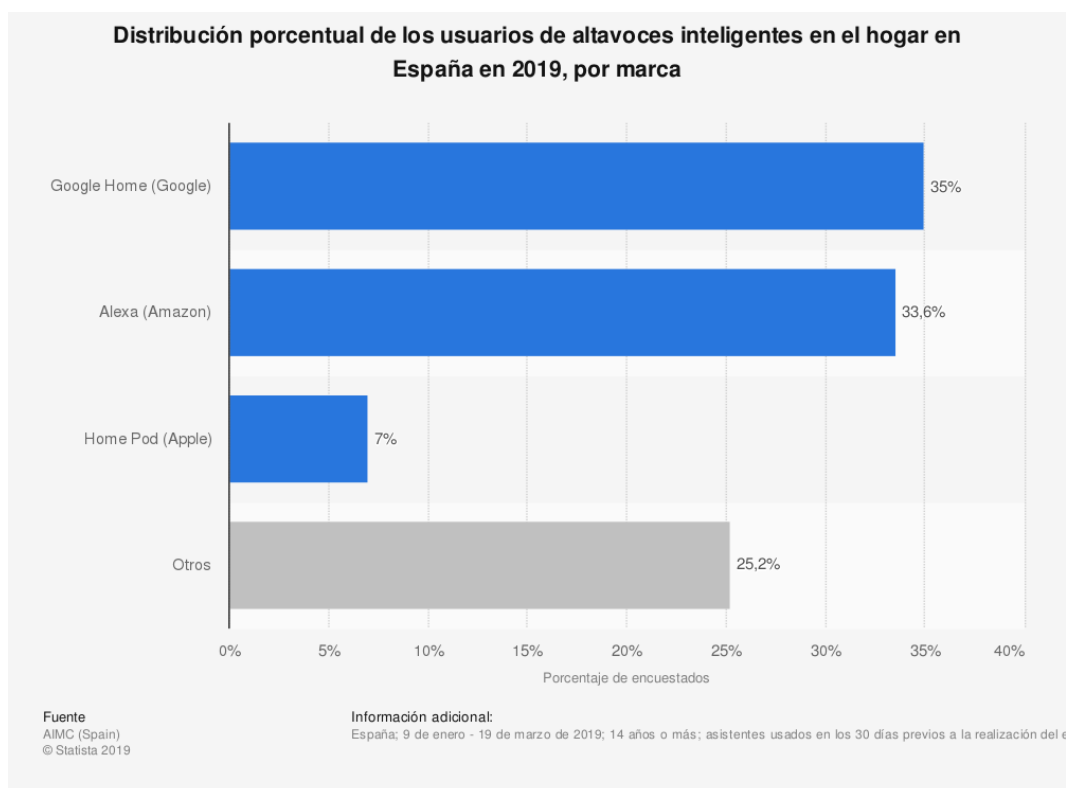


Figura 2.6: Distribución porcentual de los usuarios de altavoces inteligentes en el hogar en España en 2019, por marca. [13]

personas de su entorno [14–16].

También se ha investigado como se relacionan las personas con TEA con los asistentes virtuales y se ha llegado a la conclusión de que estas personas pueden completar algunas de las tareas que están disponibles en estos asistentes pero que todavía les falta adaptarse a las personas con diversidad funcional intelectual para que pueda ser utilizada por estas personas y que no tengan ningún tipo de problema. Lo que sí se determinó es que las peticiones que son cortas son las que resultan más fáciles de pedir a los asistentes virtuales por estas personas [1]. En uno de los estudios incluso se desarrolla funcionalidad adicional en un dispositivo Amazon Echo en el que se utiliza la voz para obtener recursos de vídeo que las personas con TEA querían ver en una tableta. En este estudio, se llega a la conclusión de que existen situaciones en las cuáles los niños con TEA no consiguen terminar algunas tareas completamente debido a sus problemas de comunicación pero que sí completan la mayoría de ellas (aquellos que pueden hablar). También se llega a la conclusión en este estudio de que al desarrollar la funcionalidad adicional al asistente adaptada a las personas con TEA mejora mucho la tasa de éxito al completar la tarea que se realizaba antes sin la funcionalidad desarrollada mediante la personalización del lenguaje a uno cercano a los sujetos que eran objeto de estudio [17].

Por otro lado, también es interesante que se han utilizado asistentes personales (de voz, de imagen y de vídeo) para incrementar la independencia en la realización de tareas de manera autónoma por parte de personas con TEA. Sería interesante el insertar un asistente virtual de voz como los que existen adaptado a que, a parte de la voz, se puedan soportar imágenes y vídeos para hacer estos asistentes virtuales más accesibles [18].

Existe también una iniciativa en la cuál se ha diseñado ropa inteligente terapéutica llamada **Tell Me** en la cuál las personas con TEA hablan mediante un micrófono integrado en la ropa (o se activa la funcionalidad mediante sensores integrados que detecten una actividad inusual en las funciones vitales de la persona) y puede aprender de forma natural mediante la voz como expresar sus sentimientos, emociones y opiniones para la situación emocional en la que se encuentre la persona [19].

2.5. Conclusiones

Teniendo en cuenta aquello que se ha presentado anteriormente se puede ver que los asistentes virtuales están tomando una mayor relevancia en el mercado internacional desde hace unos pocos años gracias a que se están comercializando en altavoces inteligentes. Otro motivo es que cada vez se están desarrollando más soluciones del Internet de las Cosas integradas con asistentes como Alexa y Google Assistant. Este hecho junto con el de que los asistentes virtuales están cada vez en más hogares hacen que se deba explorar en el ámbito de desarrollos para personas con TEA.

Además se ha visto anteriormente que no hay una gran variedad de soluciones en los asistentes virtuales para las personas con TEA. Este hecho, aumenta la importancia de investigar en esta

tecnología y brindar la posibilidad de ayudar a estas personas por medio de estos nuevos dispositivos.

El altavoz de Amazon, es con diferencia, uno de los mejores altavoces inteligentes dado que es de los que más se vende y se puede extender la funcionalidad del mismo mediante un desarrollo de terceros. También es un altavoz que se va mejorando constantemente por parte de las características implementadas de base por el asistente. La funcionalidad que puede utilizarse para el desarrollo de funcionalidad de terceros cada vez es mayor y se aumenta con relativa rapidez.

Por los motivos anteriormente dados se ha elegido desarrollar funcionalidad para personas con autismo en asistentes virtuales (dentro de altavoces inteligentes). En especial, se ha elegido Alexa ya que su altavoz ha sido ampliamente vendido mundialmente con bastante diferencia de los demás y posee un SDK con mucha funcionalidad disponible para desarrollar. Además, se ha elegido este producto debido a que por su diseño y concepto tiene muchas posibilidades de mantenerse como el líder de los altavoces inteligentes en un futuro (a no ser que aparezca un nuevo altavoz inteligente con un mejor diseño software).

DISEÑO

3.1. Necesidades de las personas con TEA

Según lo adelantado anteriormente, las personas con TEA tienen grandes déficits respecto a la interacción social y la comunicación con otros individuos. Este hecho se puede extender a los asistentes virtuales dado que la interacción que los individuos tienen con los mismos se asemeja mucho a la interacción con una persona. Por ello aquella tecnología que utilicemos se debe adaptar para que dentro del marco de la comunicación entre el asistente virtual y la persona con TEA sea accesible y tanto la persona pueda entender aquello que el asistente virtual responde como que se utilice un lenguaje que las personas con TEA puedan utilizar fácilmente para programar tareas en el asistente virtual.

Enlazando con lo anterior, las personas con TEA tienen unas capacidades comunicativas y sociales limitadas. Las personas con TEA tienen patrones repetitivos de comportamiento, de actividades e intereses que se pueden externalizar en síntomas como obsesiones, rebeldía y frustración. Este comportamiento lleva a que los educadores de las personas con TEA puedan asociar ciertos estímulos que le proporcionan a esa persona con un patrón de acciones que deban realizar (mediante el entrenamiento de pruebas discretas). Por ello, existe la necesidad de utilizar unos estímulos de respuesta que sean lo más flexibles y variados posibles para que puedan facilitar la labor del educador en cuanto a la configuración de las mismas según las circunstancias específicas de la persona o grupo de personas a las que está educando. Es necesario que se pueda personalizar el estímulo dado que el educador es el que debe decidir qué estímulo es más efectivo y entiende mejor a la persona y cómo reacciona ante diferentes estímulos. Entonces necesita tener una amplia elección para que pueda elegir qué estímulos son los más adecuados para desencadenar los patrones de comportamiento aprendidos por la persona con TEA que el educador quiere que se manifiesten.

Debido a la disfunción ejecutiva de estas personas y el problema de que no puedan controlar bien los impulsos y respuestas inapropiadas se debe delegar en un educador que elija los estímulos que menos perturben a la persona con TEA para evitar la manifestación de comportamientos y respuestas indeseadas.

3.2. Requisitos del proyecto

Se ha visto en puntos anteriores que el objetivo prioritario del desarrollo que se va a realizar será que las personas con TEA puedan interactuar más fácilmente con los asistentes virtuales. En este caso sería el facilitar la interacción de personas con TEA con asistentes virtuales para completar la tarea de poner un temporizador y que estas personas entiendan el estímulo de respuesta que le da el dispositivo cuando el temporizador termina.

Para elaborar los requisitos del proyecto hemos asistido a cuatro fuentes para poder elaborarlos con mayor precisión. Estas fuentes son: literatura científica, reuniones con expertos en HCI para TEA, un curso en edX sobre tecnologías de la información para TEA [20] y profesionales del centro de educación Alenta [21]. A continuación se exponen aquellos requisitos que se han obtenido partiendo de las fuentes anteriores.

3.2.1. Requisitos funcionales

RF1 La función que se debe desarrollar en el asistente virtual debe ser un temporizador.

RF2 En el temporizador se debe poder elegir el tiempo del temporizador.

RF3 El temporizador debe acabar con una respuesta previamente configurada.

RF4 En el temporizador la respuesta que se ha configurado se apagará al tiempo de activarse sin intervención externa.

RF5 Es necesario que la respuesta del temporizador se pueda configurar.

3.2.2. Requisitos no funcionales

RNF1 Crear una infraestructura de bajo coste teniendo en cuenta que ya poseemos el altavoz con el asistente virtual.

RNF2 Utilizar tecnologías que tengan un buen soporte a largo plazo.

RNF3 Hacer que no se necesiten de conocimientos técnicos en informática para que la persona de apoyo pueda configurar fácilmente los estímulos de respuesta del temporizador.

RNF4 Utilizar un asistente virtual comercial que sea popular.

RNF5 Utilizar un asistente virtual que permita extender la funcionalidad del mismo por parte de desarrollos de terceros mediante la instalación de módulos.

RNF6 Utilizar varios estímulos para que el educador tenga una flexibilidad de respuesta ante el estímulo.

RNF7 Diseñar unos modelos de interacción con el asistente virtual para que sea más accesible para las personas con TEA.

RNF8 La configuración de la respuesta del temporizador debe realizarse en una interfaz visual.

RNF9 El temporizador debe configurarse con la voz en el asistente virtual.

3.3. Diseño de la solución propuesta

Para plantear la solución a estos requisitos se debe pensar desde un principio sobre los estímulos que utilizará el temporizador. Aquellos que se utilizarán cuando esta termine van a ser dos: luz y sonido. Después de comentar el proyecto con profesionales del colegio Alenta [21] y preguntarles qué estímulos externos se podrían utilizar concluyeron que según el estado actual de dispositivos inteligentes que se pueden controlar se podría utilizar luz y sonido que son los que mejor se adaptan a una respuesta de un temporizador.

Para poder cumplir con los requisitos descritos anteriormente se va a utilizar una infraestructura que contiene los siguientes elementos:

- Un altavoz con un asistente virtual.
- Una luz inteligente que se pueda controlar desde el asistente virtual para poder encenderla cuando termine el temporizador que se ponga en el asistente virtual.
- Una capa de accesibilidad sobre el asistente virtual que facilite la comunicación entre el asistente virtual y la persona con TEA.
- Un componente donde se pueda modificar la configuración de la respuesta del temporizador (una especie de servidor).

El altavoz elegido deberá tener la posibilidad de realizar algún desarrollo sobre el mismo para insertarle la capa de accesibilidad. Ese mismo altavoz debe ser uno de los más vendidos del mercado debido a que este proyecto está orientado a asistentes virtuales comerciales. En este altavoz se implementará la respuesta sonora y los diferentes modos de sonido que esta tendrá.

La luz inteligente debe poder controlarse electrónicamente teniendo la posibilidad de encender y apagar la luz remotamente (así como cambiar cualquier tipo de configuración). También esta luz debe poder tener diferentes modos de funcionamiento para que se puedan cambiar.

La capa de accesibilidad es muy importante que sea personalizable por la persona de apoyo y que tenga una gran variedad de modos para elegir. Así facilita que la persona de apoyo pueda elegir unos modos de respuesta del temporizador idóneos para que la persona con TEA pueda entender con facilidad los estímulos que recibe y sea fácil para ella programar las tareas.

Hay que buscar la manera de poder personalizar los estímulos de respuesta del temporizador. Para esta parte, lo más recomendable es una interfaz web en la que la persona de apoyo pueda elegir entre diferentes estímulos y pueda configurar las diferentes opciones que vienen para la luz y el sonido. Lo más fácil es tener un elemento físico dentro de la red local que actúe como servidor y desde el cuál podamos modificar las preferencias de las diferentes configuraciones del dispositivo.

La solución que se expone en este punto queda representada tal y como se ve en la figura 3.1. En esta solución la persona con TEA o la persona de apoyo pone un temporizador que dura un tiempo determinado. Estas órdenes que se realizan con la voz las captura la capa de accesibilidad que pone

el temporizador en el asistente virtual con el tiempo que se ha requerido.

Cuando termina el temporizador del asistente virtual la capa de accesibilidad activa los estímulos que se han configurado previamente y los apaga al cabo de un tiempo determinado sin que exista ningún tipo de intervención por parte de la persona con TEA o la persona de apoyo.

Los estímulos que se utilizan en la respuesta del temporizador los configura la persona de apoyo en una interfaz visual intuitiva para un usuario de informática estándar (con bajos conocimientos informáticos) que comunica a la capa de accesibilidad los estímulos elegidos y los modos en los que se utiliza cada uno. Estos estímulos externos serán activados por la capa de accesibilidad cuando el temporizador termine.

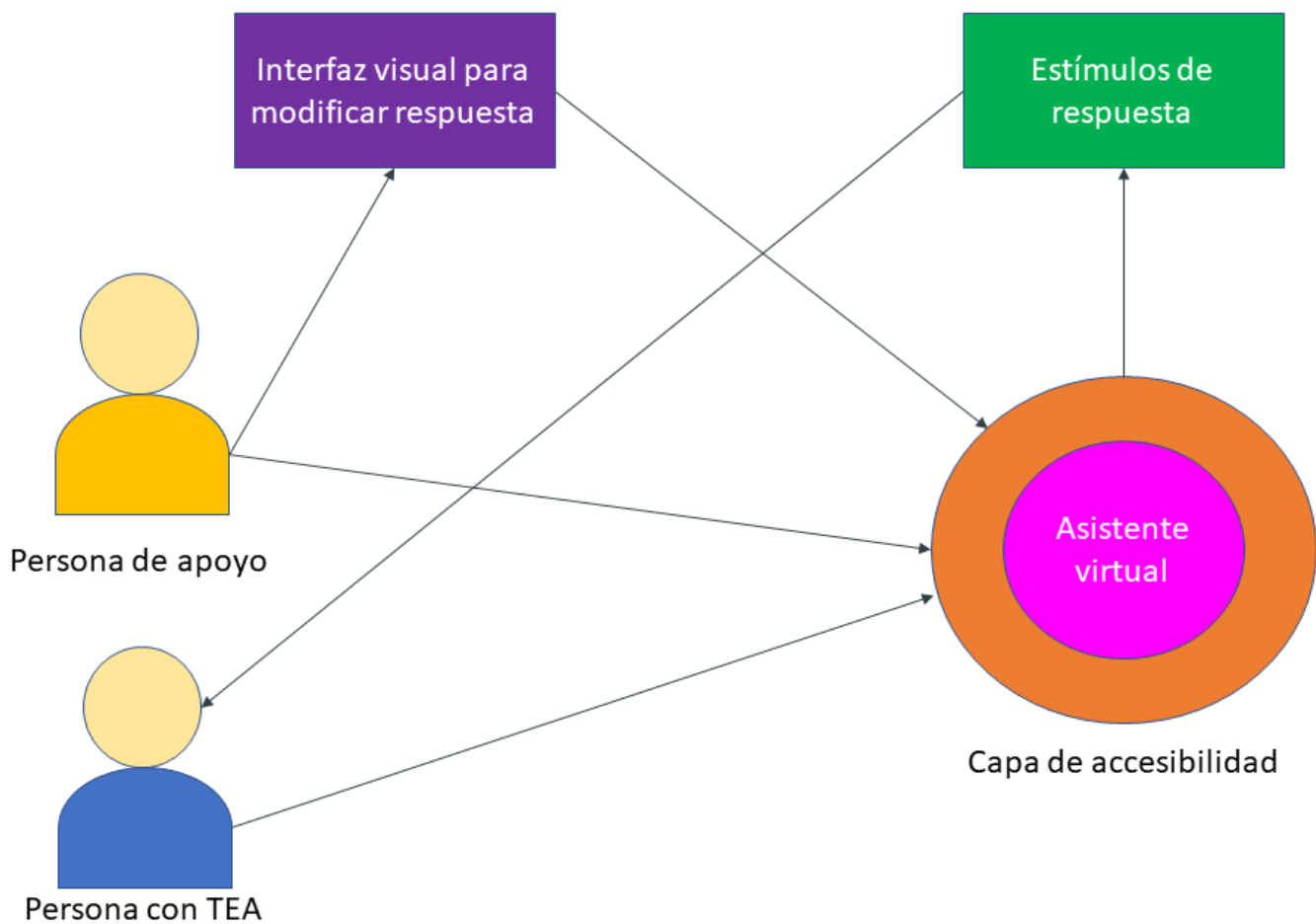


Figura 3.1: Arquitectura del diseño

IMPLEMENTACIÓN

Para poder cumplir con los requisitos descritos anteriormente primeramente se debe elegir el asistente virtual que se va a utilizar. En este caso se ha decidido utilizar Alexa como asistente virtual ya que es uno de los asistentes virtuales más vendido en el mundo (como se puede ver en la figura 2.5) y el Amazon Echo uno de los altavoces inteligentes más utilizados en España (como se puede ver en la figura 2.6) en el cuál se puede desarrollar funcionalidad de terceros para implementarla en el altavoz inteligente (una skill) dado que Google no permite que se desarrolle funcionalidad extra en sus altavoces Google Home. En la skill se desarrollará la capa de accesibilidad de la solución expresada en el diseño.

Una vez elegido el asistente virtual que se utilizará se debe desarrollar una skill en Alexa. En esta skill se debe desarrollar una parte de código en la que se activa el temporizador y se para. También se diseñará un modelo de interacción en el que se plasman las diferentes intenciones (las acciones que debe entender el altavoz) que se quieren que pueda realizar el usuario en la skill. El código se activará por intenciones. Cada intención tiene una función a la que llama cuando esta se reconoce por parte de la skill y se ejecuta para llevar a cabo las acciones que tienen que ver con esa intención. Las intenciones que reconocerá el asistente virtual son las siguientes:

- Invocación de la skill (la skill se activan como una aplicación móvil).
- Petición de ayuda: donde la skill ayudará al usuario a saber qué puede hacer con la skill.
- Petición para cerrar la skill: La skill tanto como se abre se debe cerrar.
- Petición para poner un temporizador de los minutos que quiera el usuario.

El modelo de interacción de la skill de Alexa no se puede modificar para cada persona en particular por parte del educador. Sería clave poder modificarlo para adaptar cada modelo de interacción según las palabras con que cada persona con TEA utiliza para ordenar al asistente cada una de las acciones que la skill y así se conseguiría un nivel máximo de accesibilidad. Lo que se ha decidido es crear un modelo de interacción propio que puede utilizar únicamente el educador (y aquellas personas con TEA que aprendan a hacerlo). Dada esta limitación anterior se ha planteado que la acción de poner el temporizador pueda ser realizada únicamente por el educador pero que exista una personalización del temporizador en la respuesta de la misma y que esta pueda ser variada.

En cuanto al código de la skill debe estar alojado en algún lugar. Para esto se utilizará una función Lambda de AWS ya que lo facilita mucho de cara a la gestión tanto del código como de la función. La función Lambda se encuentra integrada en la Alexa Developer Console y se puede modificar allí.

En cuanto al estímulo de la luz se decidió emplear las luces Philips Hue. Estas luces pueden ser controladas de forma remota y se pueden conectar al altavoz con el asistente virtual de Alexa (tienen soporte oficial para Amazon Alexa). Otro motivo para su elección fue que estas luces poseen distintos colores y modos de funcionamiento lo que facilitan la flexibilidad que se busca para que haya variedad de modos y colores a utilizar que puedan ser elegidos por el profesor.

Con respecto al estímulo de sonido se sabe que no se puede poner un temporizador con la API que proporciona Amazon para las skills y, por tanto, tampoco se puede cambiar el sonido de este temporizador. Además para que el educador pueda elegir una configuración de respuesta del temporizador (eligiendo los estímulos que vaya a utilizar para el temporizador) se debe establecer algún tipo de interfaz o plataforma para que pueda realizarlo. Por ello, hemos decidido utilizar una Raspberry Pi dentro de la red local donde se encuentra el altavoz para gestionar la configuración de luces y sonido y que el servidor local gestione el temporizador (dado que hemos dicho antes que no se pueden poner temporizadores desde la skill). Este componente se conectaría al altavoz con Alexa y sería utilizado como un altavoz de la Raspberry Pi. Así también se podría elegir el sonido que se quiere utilizar en el temporizador. Así, esta Raspberry Pi sería el componente adicional de esta infraestructura. Se ha elegido una Raspberry Pi debido a que es un dispositivo barato (unos 40€). Esta elección permitiría ampliar la funcionalidad en un futuro independientemente de los avances de la API de Alexa y sacar una serie de skills con funcionalidades diversas adaptadas a personas con TEA que utilizarasen el asistente virtual como intermediario.

En cuanto a la skill ésta puede enmarcarse en diferentes tipos según el modelo de interacción:

- **Custom Interaction Model:** Este modelo de interacción es libre para aquellas skills que no tienen intenciones predefinidas y tienes que definir todas aquellas intenciones que necesites que tenga tu skill.
- **Smart Home Skills:** Este modelo de interacción sirve para controlar dispositivos domóticos que sean compatibles con Alexa. En este modo sólo existen las intenciones predefinidas para este tipo.
- **Flash Briefing Skills:** Este modelo de interacción también está predefinido y sirve para dar las noticias.
- **Video Skills:** Permite dar contenido audiovisual al usuario. Este modelo de interacción está predefinido y sólo es válido si hay algún soporte visual que permita la reproducción de este tipo de contenidos.
- **Music Skills:** Este modelo de interacción está predefinido y permite proporcionar contenido de audio al usuario. También tiene una API de programación para gestionar el contenido de audio.
- **List Skills:** Este tipo de skill facilita el uso de los eventos de las listas y puede cambiar su funcionamiento o reaccionar a cambios en las mismas. No hay modelo de interacción y sólo responde a eventos producidos por las listas (añadir, modificar o eliminar un elemento de una lista).

El modelo de interacción que se definirá para la skill es el **Custom Interaction Model** porque no se ajusta a ninguno de los anteriores tipos de skill y se necesitará establecer unas intenciones propias

que utilizará el usuario para ordenar las distintas tareas del temporizador.

Para elegir la configuración de la respuesta del temporizador se debe implementar un servidor en la Raspberry Pi para hacer una interfaz web en la que se pueda elegir las diferentes opciones que podemos elegir del temporizador y la configuración actual que tiene del temporizador. Esta configuración debe estar guardada en algún lugar así que se debe implementar una base de datos en la Raspberry Pi para guardar y obtener la información.

Resumiendo todo lo que se ha hablado hasta ahora necesitamos los siguientes elementos para cumplir la funcionalidad descrita anteriormente:

- Un altavoz Amazon Echo con el que interactuar para poder programar el temporizador y que emitirá el sonido del temporizador cuando esta termine.
- Una skill en la que desarrollar el modelo de interacción entre el usuario y el dispositivo Amazon Echo.
- Una Raspberry Pi que actúe de servidor controlando el tiempo del temporizador. También activará el altavoz para que emita sonido o la luz según la configuración sobre los estímulos a utilizar que se haya realizado previamente.
- Una luz Philips Hue para utilizar cuando el estímulo elegido al terminar el temporizador sea la luz.

Los elementos se conectan de la forma que se ve en la figura 4.1:

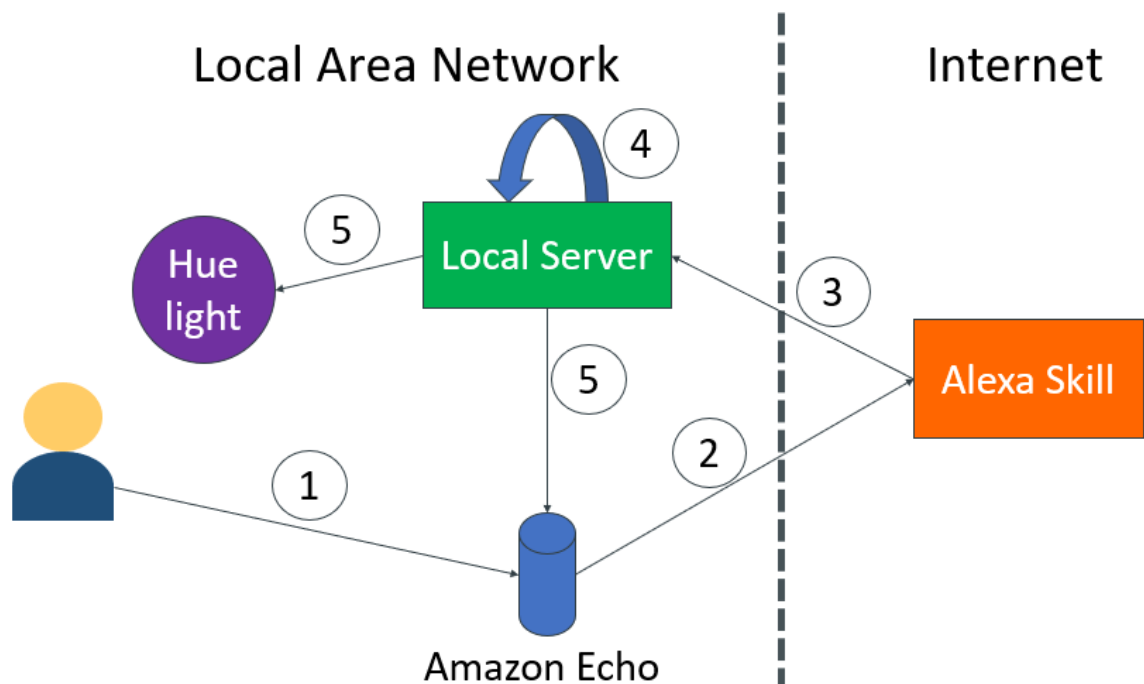


Figura 4.1: Diseño de la arquitectura y flujo de ejecución

- 1.– El usuario habla con Alexa para pedirle que ponga un temporizador que dure cierto tiempo elegido por el usuario.
- 2.– El dispositivo Amazon Echo se comunica con la skill de Alexa alojada en una Función Lambda en Amazon Web Services (AWS) que controla el flujo de la conversación del dispositivo Amazon Echo con el usuario.

- 3.– Cuando se completa el flujo de diálogo la skill de Alexa (la función Lambda) envía un mensaje mediante HTTP al servidor LAN para comenzar el temporizador con el valor de tiempo especificado por el usuario.
- 4.– El servidor espera el tiempo que le ha indicado la skill de Alexa para disparar la respuesta configurada en el mismo servidor.
- 5.– El servidor enciende finalmente la luz y reproduce el sonido en el altavoz.

4.1. Alexa Skill

La tecnología de Amazon Alexa se ha explicado anteriormente. Relativo al asistente virtual se ha elegido Amazon Alexa por diferentes razones.

Amazon Alexa es uno de los asistentes más utilizados en España y más comprados en el mundo como hemos visto antes. Esto lo convierte en una opción muy buena para elegir este asistente de voz debido a que se quiere un asistente que esté muy extendido para poder desarrollar sobre él.

Otra razón para su elección es que por medio de las skills se puede extender su funcionalidad para que pueda realizar más acciones. Además, se pueden desarrollar skills independientes e incluirlas en la tienda de skills para que todo el mundo pueda descargarla en su Amazon Alexa.

La skill es la parte de la arquitectura que se encarga de mantener un diálogo con el usuario para que éste pueda poner un temporizador. Esta parte reconoce todas las intenciones que el usuario tiene y que tienen que ver con la puesta en marcha de un temporizador.

La skill consta de dos partes:

- Modelo de interacción del usuario con la skill.
- Función Lambda en AWS que controla el flujo del diálogo y da al usuario una respuesta según la intención que el modelo de voz haya reconocido por parte del usuario.

4.1.1. Modelo de interacción.

Para poder desarrollar el modelo de interacción de la skill se debe acceder a la página web <https://developer.amazon.com>. Una vez aquí se puede crear una skill. Dentro de la consola de desarrollo de la skill hay diferentes apartados:

Build.

En esta sección se configura la skill. Los apartados a configurar son los siguientes:

- **Permisos:** En ese apartado se gestionan los permisos explícitos que se debe dar el usuario en cuanto a la recolección de datos. Datos como: correo electrónico, nombre de usuario, cuenta para realizar pagos en la skill...
- **Account Linking:** Permiso para crear una cuenta personalizada en la base de datos de la skill en AWS o

enlazarla con la cuenta de Amazon.

- **In-skill products:** Es un apartado en el que se pueden implementar funciones de pago que mejoran la experiencia o amplían funcionalidad de la skill.
- **Custom:** En este apartado se configuran elementos puramente funcionales como los siguientes:
 - **Modelo de interacción:** Esta parte es relativa a las intenciones que debe reconocer la skill y toda la interacción entre el usuario y el Amazon Echo en términos de frases que utiliza el usuario para pedir ciertas intenciones a las que la skill debe dar respuesta. Aquí se deben definir los siguientes elementos:
 - ◇ **Forma de invocación:** La frase que debe utilizar el usuario para que Alexa abra esa skill.
 - ◇ **Intenciones:** Son acciones que cumplen una acción pedida por el usuario.
 - ◇ **Entidades:** Son elementos dentro de algunas intenciones que entran dentro de una lista de valores y se necesita saber para realizar una acción.
 - **Interfaces:** Es funcionalidad adicional que se puede añadir a la skill. Esta funcionalidad añadida puede llevar a nuevas intenciones predefinidas en el modelo de interacción de la skill que hay que gestionar, establecer un flujo de diálogo y a las que dar una respuesta.
 - **Endpoint:** En este apartado se configura el lugar donde se está ejecutando el código de la función Lambda que controla el diálogo de la skill y que proporciona las respuestas a las intenciones reconocidas por la skill.
 - **Display:** Cuando la aplicación requiere o soporta apoyo visual en un Amazon Echo Show aquí se desarrollan las diferentes pantallas que se deben mostrar en el dispositivo.
 - **Conjuntos de anotaciones:** Son frases que se incluyen en las pruebas del modelo de interacción en las que se da la respuesta y los elementos que debería reconocer el modelo de interacción. Sirven para probar la precisión del modelo y evaluarlo para ver si está bien diseñado.
 - **Historial de intenciones:** Aquí salen las frases utilizadas por los usuarios que no han sido reconocidas por ninguna intención y las que sí lo han sido con el porcentaje de confianza, la acción realizada de respuesta por parte del asistente y las entidades reconocidas.

Code.

Esta parte contiene el código referente a la skill. Esta parte tiene un endpoint público creado en AWS por la Alexa Developer Console al crearse la skill. El código de la skill se puede modificar aquí.

Test.

En esta parte se puede simular una interacción real entre un usuario y la skill. En ella se pueden escribir las frases que diría el usuario y se pueden probar las respuestas que da la skill para poder saber si se han desarrollado la skill correctamente.

Distribution.

En este área se debe completar la información mostrada al usuario en la tienda de Amazon antes de descargar la skill en su dispositivo. Se puede encontrar entre la misma:

- Nombre de la skill.
- Descripción de la funcionalidad.
- Funcionalidad nueva añadida (en caso de actualización).
- Frases de ejemplo.
- Icono de la skill.

Entre otra información.

Certification.

En este área se valida que se hallan realizado todas las acciones necesarias para la publicación de la skill y se envía la skill para que el equipo de Amazon la analice y la revise para poder publicarla en la tienda de Amazon y que ésta pueda ser descargada por el público en general de las regiones que elijamos distribuirla. En nuestro caso únicamente sería en España.

Analytics.

En esta parte se pueden ver estadísticas de la skill. Las estadísticas a las que se tiene acceso son relativas a la actividad de los usuarios con la skill. Algunas de estas actividades pueden ser: número de invocaciones, número de descargas, porcentaje de respuestas exitosas, latencias, frases utilizadas para activar la skill o reconocer las intenciones, latencia, flujo de diálogo seguido...

Estas estadísticas son interesantes para detectar problemas sobre la invocación, la detección de intenciones o errores en la respuesta de la skill. De esta forma podemos modificar la skill para incluir nuevas intenciones que se detecten como necesarias, eliminar las que no son necesarias, mejorar los flujos de diálogo y las frases de identificación de intenciones. Para resumirlo, estas estadísticas permiten mejorar la experiencia del usuario con la skill.

4.1.2. Función Lambda en AWS.

La parte de la skill que gestiona la conversación entre el usuario y Alexa y aquella que proporciona las respuestas al usuario realizando las acciones necesarias para completar las órdenes que el usuario realice debe estar expuesta en Internet. Además, el servicio debe estar activo 24/7 ya que todas las peticiones de los diferentes usuarios de la skill ejecutarán este código (y puede ser simultáneo).

Amazon tiene integrado este servicio dentro de la consola de desarrollo de Alexa. Por este motivo, se utilizarán las Funciones Lambda para alojar el código de la skill.

Para almacenar el código de la skill se deben realizar los siguientes pasos:

- 1.— Se entra en <https://developer.amazon.com/> (Accedido: 02-06-2020) y se inicia sesión para entrar en la Alexa Developer Console.

- 2.— Se presiona el botón Create Skill.
- 3.— Se elige un nombre para la skill, el lenguaje por defecto, el modelo a añadir en la skill (que en nuestro caso es Custom) y el método de alojamiento para el servidor de backend (en nuestro caso, Alexa-Hosted (Node.js) dado que se va a desarrollar la skill en Javascript).
- 4.— Se elige la plantilla en JavaScript (que será Hello World Skill) y se presiona el botón Choose.
- 5.— Se rellena el captcha y esperamos un momento.
- 6.— De esta forma ya se ha creado la skill con su servicio de *backend*.

De esta manera vemos que es muy fácil crear una skill con su servicio de *backend*. El desarrollador para lo básico puede limitarse a programar y definir el modelo de interacción sin preocuparse por configuraciones extra en la consola de AWS para la función Lambda. Para funciones más avanzadas que requieran de elementos de AWS si se necesita un conocimiento de gestión de la consola de AWS.

4.2. Altavoz Echo

El Amazon Echo es un altavoz inteligente desarrollado y vendido por Amazon con funciones, que incluye la respuesta a las preguntas y el reproductor de música.

Este dispositivo incluye un procesador Texas Instruments DM3725 ARM Cortex-A8, 256MB de LPDDR1 RAM y un espacio de almacenamiento de 4GB. La conectividad se realiza por doble banda Wi-Fi 802.11a/b/g/n y Bluetooth 4. El Echo debe estar conectado siempre a la corriente para funcionar.

El Amazon Echo es un dispositivo que tiene Alexa como asistente virtual incluido en su interior. Se ha utilizado este dispositivo ya que es un altavoz con asistente virtual y es el centro neurálgico del proyecto. Además, se ha elegido este dispositivo porque es el más vendido en el mundo actualmente como decíamos anteriormente.

Las facilidades de este dispositivo es que se pueden conectar dispositivos para reproducir sonidos por Bluetooth mientras tenemos la funcionalidad de Alexa (en cuanto a que podemos interaccionar con el dispositivo mientras el dispositivo está conectado). De esta manera, se obtiene un dispositivo independiente que funciona de manera continua como asistente virtual escuchando los sonidos del exterior.

4.2.1. Instalación de la skill en el Amazon Echo.

Para disfrutar de la funcionalidad de la skill debemos instalarla en el dispositivo como si se tratase de una aplicación en un *smartphone*.

Para descargarlo en el Amazon Echo se deben realizar los siguientes pasos:

- 1.— Hay que entrar en la *app* Alexa en la cuál esté registrado el dispositivo Amazon Echo.

- 2.— Se presiona el botón del menú y se eligen **Skills y juegos**.
- 3.— Se presiona en la lupa y se busca la skill.
- 4.— Se entra en el resultado donde aparezca la skill y se presiona el botón **PERMITIR SU USO** para instalarla.
- 5.— Se esperan unos momentos y cuando cambie el botón a **DESACTIVAR SKILL** ya estará instalada en el Amazon Echo y se puede utilizar.

4.3. Servidor Raspberry Pi

El servidor utilizado se trata de una Raspberry Pi. La Raspberry Pi es un ordenador de pequeñas dimensiones. Es de tipo Single Board Computer (SBC) de bajo coste y desarrollado por la Raspberry Pi Foundation. El objetivo de este ordenador es aumentar la enseñanza de informática en las escuelas y se ha popularizado también en la robótica.

El servidor local del que se habla se trata de una Raspberry Pi versión 3 en el que se ha implementado un programa que actúa como servidor HTTP. Este servidor recibe las peticiones para iniciar el temporizador por parte de la función Lambda y tiene la interfaz de configuración de la respuesta del temporizador.

Se ha elegido este mini ordenador debido a que sería un elemento independiente de toda la arquitectura que se podría automatizar para que realizase las tareas de mantener el temporizador y todas las que permitiese hacer la skill permitiendo que se pueda adquirir un elemento de estas características aparte. Se ha elegido también este elemento para que su instalación sea fija y no haga falta moverlo una vez instalado. Aparte este elemento es de un coste reducido.

4.3.1. Programa servidor.

Node.js es un entorno multiplataforma, de código abierto, como fin en la capa del servidor basado en el lenguaje de programación **ECMAScript** (JavaScript). Este entorno es asíncrono, con funciones de entrada y salida de datos en una arquitectura que se orienta a los eventos. Este entorno se creó para ser útil en la creación de programas de red con alta escalabilidad (por ejemplo, servidores web).

Para el desarrollo del servidor y la skill se ha utilizado Javascript con Node.js versión 13.7.0 para poder unificar el desarrollo de estas dos partes y porque es un lenguaje de programación que está más documentado en el caso de la skill que los otros dos lenguajes que ofrece (Python y Java). Javascript junto con Node.js también posee muchas facilidades para crear e implementar un servidor por medio de Frameworks.

Para la realización del programa servidor se ha utilizado el paquete npm Express.js que es un framework de aplicaciones web bastante conocido.

Interfaz de configuración.

La página de configuración es como la de la figura 4.2.

Configuración actual

Configuración

Elige como avisará el temporizador

☒ Luz

☒ Sonido

Configuración de la luz

☐ Modo alerta

Tipo de luz

☒ Colores aleatorios

☐ Color único

Configuración del sonido

☒ Audio a elegir

☐ Oración

Fichero de la alarma (solo admite .wav): Ningún archivo seleccionado

Figura 4.2: Configuración de la respuesta del temporizador

Las diferentes configuraciones que se pueden realizar para la respuesta del temporizador cuando este termina son las siguientes:

- **Luz:** Se puede configurar que se encienda la lámpara Philips Hue durante un periodo de 15 segundos. Dentro de la configuración de la luz existen las siguientes opciones:
 - **Modo alerta:** Es un modo en el cuál la luz Philips Hue se hace intermitente. Este modo se puede combinar con cualquier modo de colores.
 - **Colores aleatorios:** En este modo la luz cambia poco a poco entre varios colores.
 - **Color único:** En este modo el color es único y no cambia. Con respecto al color podemos elegir entre este modo y el anterior y ambas opciones son excluyentes (el color es único o son varios colores y aleatorios). Estos colores se pueden elegir de las siguientes formas:
 - **Colores básicos:** Se puede elegir un color de la siguiente lista: blanco, azul, rojo, ama-

rillo, verde, lila y azul celeste.

— **Control de colores avanzado:** En este control se elige el color que se quiera que tenga la lámpara en la respuesta del temporizador mediante controles RGB. EL resultado final se traduce a la escala xy. Ese valor en xy lo utiliza la lámpara Philips Hue.

- **Audio:** Se puede configurar que se reproduzca un determinado audio completamente en el Amazon Echo. Las opciones para el audio son las siguientes (son excluyentes, o se elige una o se elige la otra):
 - Un audio con terminación .wav que el usuario debe enviar al servidor previamente mediante la interfaz web de configuración.
 - Una oración que se introduce como texto que se convertirá en audio para que lo pueda reproducir el altavoz.

4.3.2. Base de datos

Tal y como se ha anticipado, las configuraciones realizadas por la persona de apoyo en la interfaz del servidor habilitada para modificarlas se guardan en una base de datos PostgreSQL.

PostgreSQL se trata de un gestor de bases de datos relacionales de código abierto (bajo la licencia PostgreSQL que es parecida a la de BSD o a la de MIT) y orientado a objetos. Este gestor también tiene el nombre de Postgres.

Para la base de datos se ha elegido el gestor de base de datos PostgreSQL 11.7. Esta tecnología de gestor de base de datos se ha elegido dado que es un gestor que se ha utilizado anteriormente en alguna asignatura de la carrera y esto conlleva que se conoce bastante bien su funcionamiento.

Estas configuraciones se guardan en una base de datos llamada **tfgalberto**, en una tabla llamada valores con los campos key y value, ambos son de tipo cadenas de caracteres. Los valores de la tabla son los que se ven en la figura 4.3. Estos valores indican:

| key | value |
|-------------|-----------------|
| luz | false |
| sonido | false |
| alerta | false |
| tipo_color | fija |
| control | basico |
| tipo_sonido | frase |
| color | blanco |
| frase | frase en blanco |
| (8 rows) | |

Figura 4.3: Tabla de valores de configuración

- **luz:** Los valores pueden ser true o false para ver si se ha elegido la opción de tener o no luz como respuesta del temporizador.
- **sonido:** Los valores pueden ser true o false para ver si se ha elegido la opción de tener o no sonido de respuesta

del temporizador.

- **alerta:** Los valores pueden ser true o false para tener el modo alerta (parpadeo intermitente de la luz) en la Philips Hue.
- **tipo_color:** Los valores pueden ser fija o colorloop. Este valor es para elegir si se quieren colores aleatorios en la luz o un color fijo.
- **control:** Este valor es para saber si el color se ha elegido con el control básico (elegir un color como tal) o con el avanzado (puedes elegir la cantidad de color rojo, verde y azul en el color final). Los valores son basico y avanzado.
- **tipo_sonido:** Los valores pueden ser audio (si es un audio a reproducir por el altavoz) o frase (si es una frase de texto la que tiene que reproducir el altavoz).
- **color:** Si elegimos que el color sea fijo aquí se detalla que color es, en la escala xy.
- **frase:** La frase que debe decir el altavoz (si es que se ha elegido esta opción).

Endpoints.

Las distintas direcciones a las que se puede acceder en el servidor son:

- **/:** En este lugar se puede acceder a la configuración de la respuesta del temporizador y configurarla a gusto de la persona de apoyo.
- **/config:** En esta página se puede acceder a la configuración actual de la respuesta del temporizador.

Cambia la configuración

Configuración actual de la alarma

Luz: No

Sonido: No

Figura 4.4: Configuración actual de la respuesta del temporizador

- **/testAudio:** Esta dirección se utiliza para probar el audio que la persona de apoyo ha subido a la página de configuración sin guardarlo en el sistema de forma permanente. Esta dirección se utiliza cuando la persona de apoyo presiona el botón de probar en la página de configuración.
- **/sendAudio:** Esta dirección se utiliza para guardar de forma permanente el audio que la persona de apoyo ha subido a la página de configuración (si éste ha elegido la opción del audio). Esta dirección se utiliza cuando la persona de apoyo guarda la configuración.
- **/sendText:** Esta dirección se utiliza para convertir el texto introducido a un audio que se reproducirá cuando el usuario el temporizador termine y guardarlo en el servidor de forma permanente. Esta dirección se utiliza cuando el usuario guarda la configuración.
- **/testText:** Esta dirección se utiliza para probar el texto que la persona de apoyo ha rellenado en el cuadro de texto de la página de configuración sin guardarlo en el sistema de forma permanente. Esta dirección se utiliza cuando el usuario presiona el botón de probar en la página de configuración.
- **/postPreferences:** Esta dirección se utiliza cuando la persona de apoyo va a guardar la configuración seleccionada en la pantalla principal. En este punto se guarda toda la configuración elegida en la pantalla principal de la base de datos.
- **/ligthOn:** Esta dirección permite probar si el servidor puede establecer una conexión con la luz Philips Hue y encender la luz.

- **/lighOff:** Esta dirección permite probar si el servidor puede establecer una conexión con la luz Philips Hue y apagar la luz.
- **/timer:** Esta dirección es a la que accede la función Lambda para poner el temporizador. Únicamente se debe pasar el número de segundos que queremos que esté el temporizador. A partir de aquí, el servidor esperará los segundos que sean necesarios y después de eso ejecutará la respuesta ajustada a la configuración establecida.

4.3.3. Conexión con el dispositivo Amazon Echo.

Se necesita que el servidor se conecte con el Amazon Echo para poder reproducir el audio del temporizador dado que la skill por sí sola no puede poner un temporizador y menos elegir el audio que esta puede tener. Al ser esta una de las opciones que necesitaba el temporizador se tenía que conectar el altavoz de alguna manera con el servidor.

Las conexiones al altavoz que pueden realizarse al Amazon Echo únicamente pueden ser mediante la tecnología Bluetooth dado que no existe otra forma de conectarse a este dentro de una red local. Por este motivo, en parte utilizamos la Raspberry Pi como servidor dado que es un ordenador de pequeñas dimensiones y de bajo coste con conexión Bluetooth.

Al intentar conectar la Raspberry Pi por Bluetooth al Amazon Echo se verifica que la vinculación y conexión entre los dos dispositivos se hacía pero la Raspberry Pi no podía mandar audio por esa vía. Al ver que no se podía conectar por esta vía, se descartó y se eligió otra opción.

La Raspberry Pi versión 3 que se utiliza tiene un puerto PCM con una conexión de miniJack 3.5 la cuál si se puede elegir como salida de audio. Entonces para poder conectar de manera remota el audio de la Raspberry que actúa como servidor con el Amazon Echo se utiliza un adaptador que convierte el sonido del miniJack 3.5 a Bluetooth. El adaptador es el que se puede ver en la figura 4.5.



Figura 4.5: Adaptador Bluetooth

Para vincular la Raspberry Pi con el Amazon Echo se deben realizar los siguientes pasos:

- 1.— Conectar el adaptador Bluetooth a la Raspberry Pi. Primero se conecta el miniJack y después se conecta al USB de la Raspberry.
- 2.— Se accede a la aplicación de Amazon Alexa. Hay que entrar en el apartado de dispositivos.
- 3.— Dentro de mis dispositivos se selecciona Echo y Alexa.
- 4.— Se elige el dispositivo Echo (MiEcho).
- 5.— Se presiona sobre el engranaje.
- 6.— Se presiona sobre dispositivos Bluetooth.
- 7.— Se presiona el botón: Enlazar un dispositivo nuevo.
- 8.— Se busca el adaptador y se pulsa.
- 9.— Se espera a que se vincule y ya se puede transmitir audio a través del dispositivo.

Después de estos pasos ya se puede reproducir el audio del temporizador (si está seleccionado) en el altavoz Amazon Echo.

4.3.4. Comunicación con la skill.

En cuanto a la comunicación entre la skill y el servidor se realizará por internet (ya que la skill sólo puede acceder a recursos públicos de Internet).

Al estar este sistema montado en una casa se han realizado las siguientes acciones:

- Establecer una dirección IP privada fija estática a la Raspberry Pi.
- Redireccionamiento de puertos para que el puerto 8080 de la dirección IP de la Raspberry Pi esté visible al exterior desde el puerto 80 de mi router.
- Reservar el dominio **tfgalberto.ddns.net** que es del DNS No-IP. Esto se ha realizado para que no haya que saber en cada momento la dirección IP pública del *router* (que además es dinámica) y utilizar un nombre de dominio para una IP que puede ser cambiante.
- En la Raspberry hay un cliente de No-IP que cada cierto tiempo actualiza la IP pública de mi *router*.

Después de todos estos pasos la skill ya se puede comunicar con el servidor y acceder a la dirección `tfgalberto.ddns.net/timer` pasándole el número de segundos del temporizador que debe configurar.

4.4. Philips Hue

Philips Hue es un sistema de iluminación inteligente que permite controlar la luz. Este sistema utiliza la domótica para controlar y automatizar las lámparas.

La respuesta lumínica se quería hacerla con un dispositivo de luz inteligente para poder modificar el estímulo y poder crear diferentes modos de ejecución modificando características como el brillo, el color, que sea o no intermitente... Además, de esta forma podíamos controlar esta luz desde el servidor

y apagarla y encenderla cuando quisiéramos.

Por estos motivos se ha elegido la Philips Hue debido a su gran capacidad de personalización y la posibilidad de control remoto.

4.4.1. Arquitectura interna de las Philips Hue.

Los elementos de la arquitectura interna de las Philips Hue son los siguientes:

- 1.– Un servidor llamado *bridge*. Figura 4.6(b).
- 2.– El terminal de luz Philips Hue. Figura 4.6(a).



(a) Luz de la Philips Hue.



(b) Bridge.

Figura 4.6: Elementos de la arquitectura de Philips Hue.

En el servidor se pueden conectar varias luces Philips Hue. Mediante este servidor se pueden gestionar las luces que controla el servidor y modificar sus estados (encender y apagar) y modos de funcionamiento (un solo color, colores aleatorios, parpadeo...). También controla que todos los accesos sean legítimos mediante un usuario que debemos crear con anterioridad y siguiendo un protocolo específico. Esta gestión se puede realizar mediante la API Rest pública documentada que se encuentra disponible para los que se den de alta en esta página web <https://developers.meethue.com/> (Accedido: 02-06-2020).

El terminal de Philips Hue es el que recibe las órdenes por parte del *bridge* y las ejecuta.

4.4.2. Conexión con el servidor.

Para realizar la conexión entre el servidor y las luces para controlarlas se deben seguir los siguientes pasos:

1.— En la primera llamada al *bridge* se debe crear un usuario para poder acceder a gestionar las luces. El protocolo para crearlo es el siguiente:

- Primero, se presiona el botón del *bridge* que permite conectarse a nuevos dispositivos.
- Se llama por el método POST al endpoint `/api` con el siguiente cuerpo:

```
{"devicetype": "my_hue_app#TFG_server_prueba"}
```

La respuesta a esta petición es una lista con el usuario que ha sido creado.

2.— Para acceder a las luces que tiene el servidor registradas se debe acceder con el método GET a la siguiente dirección: `/api/<username>/lights`. De esta manera, se obtiene en una lista todas las luces que controla el servidor Philips y se puede modificar sus características del campo *state*.

3.— Para cambiar el estado de una luz se debe acceder con el método PUT a la siguiente dirección sabiendo el id de la luz y el usuario que ha asignado el servidor: `/api/<username>/lights/<id>/state`. Con esta petición deberemos enviar un JSON en el cuerpo con los elementos del *state* que se quieren cambiar (encendido y apagado, cambio de luces, luz fija o intermitente...).

PRUEBAS

Las pruebas que se han realizado a este proyecto han sido pruebas unitarias y de integración. Se ha planteado hacer pruebas del proyecto con niños del colegio de educación especial Alenta [21] (pruebas de aceptación). Estas pruebas no se realizaron debido a la situación del estado de alarma por la COVID-19 y las medidas de distanciamiento social. Las pruebas de sistema no son necesarias ya que el sistema final es en el que se están desarrollando las pruebas unitarias y de integración.

Sobre las pruebas unitarias son unas pruebas realizadas para comprobar que cada elemento de la arquitectura funcionaba por separado para poder juntarlos todos después.

Sobre las pruebas de integración son unas pruebas que se realizan después de las unitarias para integrar todos los elementos cuando se han probado cada uno por separado y así poder probar la funcionalidad en su conjunto.

5.1. Pruebas unitarias

5.1.1. Luz

En cuanto a la luz hemos establecido unas pruebas para ver los diferentes modos que puede tener la luz. Estas pruebas son de caja blanca. Para pedir los cambios en la luz hay que realizar antes los siguientes pasos:

- 1.— Hay que localizar la dirección IP del bridge debido a que es la parte de la infraestructura de Philips que controla la luz.
- 2.— Una vez encontrada se accede a la Uniform Resource Identifier (URI) `/api` con el método POST enviando el siguiente mensaje para que cree un nuevo usuario: `{"devicetype": "my_hue_app#iphone peter"}`
- 3.— Con el nuevo usuario se puede acceder a la URI `/api/<username>/lights` para obtener todas las luces que controla el *bridge*.
- 4.— Las características de cada luz se pueden cambiar en la URI `/api/<username>/lights/<id>` utilizando el método PUT y mandando un cuerpo con un JSON con los valores nuevos de las características que queremos modificar.

Después hemos visto todas las características de configuración que podíamos cambiar que se encuentran a continuación y hemos probado todos los valores (si eran pocos) o algunos valores de referencia para probar que todas las características funcionaban correctamente (en el caso de los números).

- **on:** La luz está encendida. Valores: true/false.
- **bri:** Es el brillo de la luz. Valores: [1,254].
- **hue:** Hue de la luz. Valores: [0,65535].
- **sat:** Saturación de la luz. Valores: [0,254].
- **effect:** Efecto de la luz. Valores: colorloop/none
- **xy:** Son coordenadas de color en el estándar CIE. Valores: [0.00,1.00].
- **ct:** La temperatura de la luz. Valores: [153,500].
- **alert:** Cambio temporal del estado de la bombilla. Valores: none/select/lselect.
- **colormode:** Indica el modo de color de la luz. Valores: hs(Hue and Saturation)/xy/ct.
- **reachable:** NO se puede cambiar e indica si el *bridge* la detecta para poder cambiar su configuración. Valores: true/false.

5.1.2. Sonido

Por un lado se ha probado el audio por Bluetooth del Amazon Echo. Esto se ha realizado conectando un *smartphone* por Bluetooth con el dispositivo Amazon Echo y se ha procedido a reproducir música en el mismo. Las pruebas finalizaron con éxito.

Para probar el sonido de la Raspberry Pi se ha conectado el USB Bluetooth del que dispone al ordenador y se ha reproducido un audio que había dentro de la Raspberry Pi. Las pruebas finalizaron con éxito. Lo que si se ha detectado es que hay un ruido bajo de fondo que no se sabe si es problema del USB o de la Raspberry Pi.

5.1.3. Servidor

La parte del servidor se probará mediante peticiones HTTP al propio servidor para probar que funcionan correctamente todas las direcciones del mismo.

Las pruebas que se realizarán serán de caja negra. Probaremos que todas las peticiones respondan correctamente. Hay algunas que no podemos hacerlo debido a que utilizan elementos con los que se deben integrar.

Las peticiones que realizaremos serán las que se ven a continuación.

- **/:** Únicamente se hace una petición GET para ver si se muestra la página donde configurar la respuesta del temporizador.

- **/config**: Únicamente se hace una petición GET para ver si se muestra la página donde nos dice lo que se ha configurado en el dispositivo como respuesta.
- **/testAudio**: NO se probará porque se necesita la integración con el Amazon Echo.
- **/sendAudio**: En esta parte se probará a enviar un audio y se comprobará que se ha guardado en la Raspberry Pi. Dado que este método se utiliza en la interfaz de configuración del temporizador la propia interfaz controla que se envía siempre un audio cuando se manda a esta dirección.
- **/sendText**: En esta parte se probará a enviar un texto y se comprobará que se ha guardado el audio que contiene el *speech* del texto enviado en la Raspberry Pi. Dado que este método se utiliza en la interfaz de configuración del temporizador, la propia interfaz controla que se envía siempre un texto cuando se manda a esta dirección.
- **/testText**: NO se probará porque se necesita la integración con el Amazon Echo.
- **/postPreferences**: Se hará una petición POST para comprobar si se modifican los valores que se indican.
- **/lighthOn**: NO se probará porque se necesita la integración con el Philips Hue.
- **/lighthOff**: NO se probará porque se necesita la integración con el Philips Hue.
- **/timer**: Comprobaremos que con una petición POST pone un temporizador con el número de segundos que ha pasado. Dado que este método se utiliza en la skill la propia skill controla que se envía siempre un número positivo de segundos cuando se manda a esta dirección.

5.1.4. Skill

En cuanto a las pruebas realizadas en la skill únicamente se hicieron de caja negra dado que no se puede realizar ningún tipo de depuración sobre el código de la skill. Por ello, lo que probaremos en esta parte serán las respuestas a las órdenes que le damos (probaremos el modelo de interacción) y las intenciones que reconoce. Las órdenes utilizadas se encuentran en la siguiente lista con su respectiva intención:

- **SetAlarmIntent**: quiero poner una alarma, quiero poner una alarma de 5 minutos, una alarma de 5 minutos, una alarma, un temporizador de 3 minutos.
- **CancelIntent**: Cancélalo, no quiero poner la alarma, cancela todo.
- **HelpIntent**: qué hago, me ayudas, no se qué hacer, ayúdame, quiero ayuda, ayuda.
- **StopIntent**: quiero que pares, cállate, para por favor.
- **NavigateHomeIntent**: quiero ir fuera, quiero salir.

5.2. Pruebas de integración

5.2.1. Luz

En cuanto a la integración de la luz con el servidor local utilizaremos uno de los usuarios creados anteriormente en el *bridge* para modificar las características de las luces. Para ver la correcta integración de la luz Philips Hue con el servidor llamaremos a las URI s del servidor **/lighthOn** y **/lighthOff** para poder probar que la luz se enciende y se apaga correctamente en cada caso.

5.2.2. Sonido

Para probar el sonido lo que se debía hacer en una primera instancia era conectar la Raspberry Pi con el USB Bluetooth al Amazon Echo como hemos explicado anteriormente. Después de este paso se probó reproducir un audio mp3 en la Raspberry Pi y se escuchaba en el Amazon Echo.

5.2.3. Servidor

Para probar el servidor ahora probaremos aquellas URI s que dependan de un elemento externo al servidor (la luz o el Amazon Echo). Las URI s a probar son las que se muestran en la tabla ???. En estas pruebas de integración del servidor no hace falta comprobar si está conectado el audio al Amazon Echo o a la luz Philips Hue porque los errores que se producen en ausencia de estar conectados no son fatales.

- **/testAudio:** En esta parte se probará a enviar un audio y se comprobará que se ha guardado en la Raspberry Pi. Justo después de guardarse este audio debe reproducirse en el Amazon Echo. Dado que este método se utiliza en la interfaz de configuración del temporizador la propia interfaz controla que se envía siempre un audio cuando se manda a esta dirección.
- **/testText:** En esta parte se probará a enviar un texto y se comprobará que se ha guardado el audio que contiene el *speech* del texto enviado en la Raspberry Pi. Después de guardarse ese audio se debe reproducir en el Amazon Echo instantáneamente. Dado que este método se utiliza en la interfaz de configuración del temporizador la propia interfaz controla que se envía siempre un texto cuando se manda a esta dirección.
- **/lighthOn:** En esta dirección se probará si se puede encender la luz de las Philips Hue mandando una orden de modificación del *state* poniendo on al valor true.
- **/lighthOff:** En esta dirección se probará si se puede apagar la luz de las Philips Hue mandando una orden de modificación del *state* poniendo on al valor false.
- **/timer:** Comprobaremos que con una petición POST pone un temporizador con el número de segundos que ha pasado y ejecuta el programa configurado en la base de datos de la Raspberry Pi. Así veremos si se integra bien el servidor con el Amazon Echo y la luz Philips Hue y cumple bien su función.

5.2.4. Skill

Para comprobar la integración de la skill con los demás elementos de la arquitectura tenemos que configurar la intención **SetAlarmIntent** para que haga una petición HTTP al servidor local (cuya Uniform Resource Locator (URL) es: `http://tfgalberto.ddns.net`) a la URI **/timer**. Si al poner el temporizador en el tiempo que deseamos esta reproduce el programa que hemos configurado en el servidor sabremos que todo el sistema funciona correctamente. EL paso de probar la integración de la skill debe ser el último.

CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones

Una vez acabado el proyecto se puede concluir que existen algunas limitaciones en cuanto al desarrollo de funcionalidad de terceros en asistentes virtuales (concretamente Alexa) para brindar accesibilidad a las personas con TEA.

Una de las limitaciones es que el modelo de interacción por voz debe seguir un patrón en el cuál primero se debe invocar la skill y después se debe pedir a la skill lo que se desea que se haga. Este proceso de que haya dos intervenciones para poder realizar la acción que se desea puede producir que la persona con TEA no pueda llegar a completar correctamente la acción que desea ordenar ya que tienen problemas por un desarrollo del lenguaje más reducido. Están probando una función que es que se puedan pedir acciones a skills sin tener que realizar la invocación que ahora mismo esta en fase beta y se está probando con un grupo reducido de desarrolladores. Esperemos que esta opción esté disponible en un futuro.

Otra limitación en Alexa es que no se puede modificar el modelo de interacción de la skill para que la persona de apoyo pueda configurar las acciones con las palabras que mejor se adaptan a la persona con TEA. Este modelo de interacción es definido por el desarrollador y es el mismo para todos los dispositivos en los que se instala la skill.

Una última limitación de estos sistemas es que existe una limitación en las funciones que se pueden realizar internamente en el dispositivo con Alexa sobre controlar dispositivos domóticos enlazados con el dispositivo, poner temporizadores y dejar la skill en segundo plano (ahora mismo se pueden realizar funciones de mantener una conversación con el dispositivo, enviar notificaciones, recabar información de Internet para proporcionársela a la persona con TEA por la voz o realizar peticiones en Internet). Algunas de estas funciones se están integrando actualmente pero han lastrado un poco el desarrollo de la skill teniendo que externalizar esa funcionalidad al servidor externo en la red local.

También existe la ventaja de que los asistentes virtuales tienen un modo diferente de interacción con la persona con TEA que las herramientas que se utilizan para personas con diversidad funcional.

Estas soluciones pueden ser útiles para personas con diversidad funcional intelectual y de movilidad dado que las acciones se piden por la voz que es más accesible que la escritura y no necesita de una interacción explícita con el dispositivo (al interactuar con la voz). Esto puede servir para que trabajos futuros se realicen centrándose en personas con movilidad reducida, con dificultades de visión y otras diversidades.

Otra de las cosas positivas es que se ha podido solucionar los problemas de la utilización del temporizador y el dispositivo domótico mediante el servidor en la red local en el que cambiamos la configuración del temporizador (creando una arquitectura de bajo coste).

Al final se ha conseguido desarrollar funcionalidad extra para un asistente de voz que ha facilitado que los usuarios con TEA puedan comprender un temporizador. Esto lo hemos logrado gracias a que hemos integrado una variedad de estímulos y configuraciones disponibles para el temporizador.

También se ha conseguido desarrollar un servicio para personas con TEA y así aumentar la oferta de servicios disponibles que tienen estas personas en el asistente.

Por otro lado, el objetivo de ser fácil de configurar se ha cumplido con el desarrollo de una interfaz web para la configuración de los estímulos que se dan al finalizar el temporizador.

Finalmente, se ha podido ver que existen ciertas limitaciones y se han acotado para tomarlas en cuenta en desarrollos futuros aunque los asistentes virtuales están preparados para implementar soluciones para personas con TEA. Así, este trabajo puede ser el punto de partida para poder solucionar fallos cometidos en el mismo y ver la evolución de las opciones de desarrollo de los diferentes asistentes virtuales del mercado para poder mejorar las soluciones desarrolladas para personas con diversidad funcional en un futuro.

6.2. Trabajo futuro

Con respecto al trabajo futuro de este proyecto deben establecerse en un primer momento pruebas de la skill con personas con TEA para verificar en un entorno real los resultados que produce este proyecto. Así se pueden ver los fallos que tiene la skill para arreglarlos y ver también los puntos fuertes que tiene.

Dada la versatilidad de funciones que se pueden desarrollar en la arquitectura que se presenta se deja como trabajo futuro ampliar la funcionalidad de la skill para poder ampliar la accesibilidad a estas tecnologías y ayudar a que las personas con TEA puedan utilizar los asistentes virtuales con mayor facilidad.

Por último, el proyecto se iba a presentar al XXI CONGRESO INTERNACIONAL DE INTERACCIÓN PERSONA-ORDENADOR [22] que se realizaba en septiembre de 2020 pero por la situación del COVID-19 se va a posponer a Septiembre del 2021. Queda como trabajo futuro enviarlo al congreso.

BIBLIOGRAFÍA

- [1] L. Torres Valverde, “Estudio de asistentes virtuales en el entorno de la diversidad funcional intelectual,” 2019.
- [2] R. Goodman, *Child and adolescent psychiatry*. Chichester, West Sussex: John Wiley & Sons, 3rd ed. ed., 2012.
- [3] “Definition: intellectual disability,” apr 2020.
- [4] C. McDougle, *Autism Spectrum Disorder*. Oxford, UK: Oxford University Press, 02 2016.
- [5] *Diagnostic and statistical manual of mental disorders : DSM-5*. Washington DC, London: American Psychiatric Publishing, 5th ed. ed., 2013.
- [6] E. L. Hill, *Evaluating the theory of executive dysfunction in autism*. American Psychiatric Publishing, 2006.
- [7] C. Hughes, J. Russell, and T. W. Robbins, “Evidence for executive dysfunction in autism,” *Neuropsychologia*, vol. 32, no. 4, pp. 477 – 492, 1994.
- [8] G. Moreno, “ Gráfico: Apple se apunta al mercado de los altavoces inteligentes | Statista.” <https://es.statista.com/grafico/9676/apple-se-apunta-al-mercado-de-los-altavoces-inteligentes/>, jun 2017. Accedido: 02-06-2020.
- [9] R. Fernández, “ Asistentes virtuales de voz más usados en España 2019 | Statista.” <https://es.statista.com/estadisticas/1012695/asistentes-virtuales-de-voz-mas-usados-en-espana/>, jun 2019. Accedido: 02-06-2020.
- [10] R. Fernández, “ Asistentes virtuales en uso en el mundo 2019-2023 | Statista.” <https://es.statista.com/estadisticas/972995/asistentes-virtuales-en-uso-en-el-mundo/>, feb 2019. Accedido: 02-06-2020.
- [11] R. Fernández, “ Altavoces inteligentes: cuota de mercado de fabricantes 2017-2018 | Statista.” <https://es.statista.com/estadisticas/586164/cuota-de-mercado-mundial-de-los-fabricantes-de-altavoces-inteligentes/>, jun 2019. Accedido: 02-06-2020.
- [12] G. Moreno, “ Gráfico: El mercado de altavoces inteligentes se pone interesante | Statista.” <https://es.statista.com/grafico/15161/envio-de-altavoces-inteligentes/>, aug 2018. Accedido: 02-06-2020.
- [13] R. Fernández, “ Altavoces inteligentes: usuarios por marca España 2019 | Statista.” <https://es.statista.com/estadisticas/1012720/altavoces-inteligentes-porcentaje-de-usuarios-por-marca-en-espana/>, jun 2019. Accedido: 02-06-2020.

- [14] M. L. Olive, B. De La Cruz, T. N. Davis, J. M. Chan, R. B. Lang, M. F. O'Reilly, and S. M. Dickson, "The effects of enhanced milieu teaching and a voice output communication aid on the requesting of three children with autism," *Journal of Autism and Developmental Disorders*, vol. 37, pp. 1505–1513, sep 2007.
- [15] M. M. Schepis, D. H. Reid, M. M. Behrmann, and K. A. Sutton, "INCREASING COMMUNICATIVE INTERACTIONS OF YOUNG CHILDREN WITH AUTISM USING A VOICE OUTPUT COMMUNICATION AID AND NATURALISTIC TEACHING," *Journal of Applied Behavior Analysis*, vol. 31, pp. 561–578, dec 1998.
- [16] I. Torii, K. Ohtani, N. Shirahama, T. Niwa, and N. Ishii, "Voice output communication aid application for personal digital assistant for autistic children," in *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, pp. 329–333, 2012.
- [17] A. A. Allen, H. C. Shane, and R. W. Schlosser, "The Echo™ as a Speaker-Independent Speech Recognition Device to Support Children with Autism: an Exploratory Study," *Advances in Neurodevelopmental Disorders*, vol. 2, pp. 69–74, mar 2018.
- [18] L. C. Mechling, D. L. Gast, and N. H. Seid, "Using a personal digital assistant to increase independent task completion by students with autism spectrum disorder," *Journal of Autism and Developmental Disorders*, vol. 39, pp. 1420–1434, may 2009.
- [19] H. Koo, "'tellme': Therapeutic clothing for children with autism spectrum disorder (asd) in daily life," in *Proceedings of the 2014 ACM International Symposium on Wearable Computers: Adjunct Program*, ISWC '14 Adjunct, (New York, NY, USA), p. 55–58, Association for Computing Machinery, 2014.
- [20] "Aplicaciones móviles y otras tecnologías para personas con tea." https://edge.edx.org/courses/course-v1:UAM_FO+ApMovTEA+2019_T3/course/. Accedido: 02-06-2020.
- [21] "Alenta." <https://www.alenta.org/colegio>. Accedido: 02-06-2020.
- [22] "XXI congreso internacional de interacción persona-ordenador." <https://www.interaccion2020.uma.es/>, 2020. Accedido: 02-06-2020.

DEFINICIONES

skills Se trata de una funcionalidad extra desarrollada por un tercero que se puede instalar en los distintos asistentes virtuales por voz que tienen Alexa instalado. En una analogía con los dispositivos móviles las skills se podrían considerar las aplicaciones de los dispositivos móviles. Las skills son de diferentes categorías y, al activarlas se encuentran disponibles en todos los dispositivos Alexa vinculados a la cuenta de Amazon.

ACRÓNIMOS

AWS Amazon Web Services.

HCI Human-Computer Interaction.

OMS Organización Mundial de la Salud.

SBC Single Board Computer.

SDK Software Development Kit.

SO Sistema Operativo.

TEA Trastornos del Espectro Autista.

TFG Trabajo Fin de Grado.

URI Uniform Resource Identifier.

URL Uniform Resource Locator.

USB Universal Serial Bus.

